1992 / AUGUSZTUS

ÁRA: 196 FT

ALAPLAP



MIKROSZÁMÍTÓGÉP MAGAZIN MÁGNESLEMEZ MELLÉKLETTEL

A HÓNAP TÉMÁJA:

SZÓCSÉPLÉS?

Gépírók a mennyben

Csaknem Basic

Mi muzsikus lelkek...

Számkígyó-bűvölés

Volt egyszer egy PC Turbo Klub

Intelligencia minden szinten

A MÁGNESLEMEZEN:

Komplex számolás
„Biliárdozó marslakók"
Tilitoli Herculesre is
Szóhossz-statisztika
Számfelismerősdi

Taszkkommunikációs eszközök

A hardver(es) jövője

A CorelDRAW méltó vetélytársa

Átlépés egy másik dimenzióba



1. Támasztékos
Könnyű, könnyedén mozgatható. A legolcsóbb modell. (Ezen nincs digitális óra.)

PerfectData®

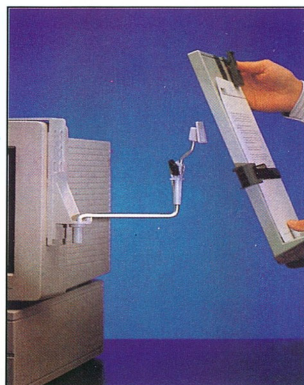
Kézirattartók — kívánság szerint



2. Talpas
A legnépszerűbb változat. Bármilyen asztalfelületen könnyedén, kis helyen lerakható, máshová bármikor áttehető.



3. Alátétes
Kis helyet foglal el. A számítógép, a monitor vagy az írógép alá csúsztatott támaszték hordozza az irattartó táblát.



4. Felragasztható
Kétoldalas tapadószalaggal rögzíthető a monitor oldalára. Asztalfelületet nem igényel. Használaton kívül a tábla leemelhető róla.



5. Hosszúkarú
Nagy teherbírású, sokféleképpen beállítható, az asztallapra szorítócsavarral felerősíthető.

Általános szolgáltatások:

— Beépített digitális órával felszerelt papírcsíptető.
— A kéziratok vastagságához igazodó, átlátszó sorvezető.
— A nagyobb méretű eredetiket is rögzítő szerkezet.
— Állítható dőlésszögű tábla.

Az igényesebbeknek néhány extra lehetőség:

— Lábpedálos működtetésű, motorizált sorvezető. Ezáltal a gépelés megszakítása nélkül, folyamatosan követhető az olvasott szöveg.
— Olvasólámpa a táblára erősítve. Különösen a nehezen olvasható kéziratokhoz hasznos.
— Nagyító sorvezető az apróbetűs szövegekhez. Egyszerre két normál sornyi tekinthető át vele.
— Táblatoldalék a szélesebb kéziratokhoz.

Cédrus Karolina Áruház
Budapest XI., Karolina út 17. Tel.: 166-2111 • Fax: 185-2221



ALAPLAP

Mikroszámítógép magazin
mágneslemez melléklettel

Megjelenik havonta

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

Munkatárs:
Sziebig Andrea

A mágneslemez melléklet,
a Lemezkalauz
és a Közkincs szerkesztője:
Verebély Pálné

A szerkesztőbizottság tagjai:
Barna László, Boros György,
Broczkó Péter, Brüll Károly,
Farkas Ernő, Feleki Zoltán,
Herczeg József, Kassay Árpád,
Kónya László, Kovács P. Attila,
Pintér Gábor, Vargha Dénes,
Vékony Tamás, Villányi László,
Zoltai Péter

Szerkesztőség, kiadó
és hirdetésszervezés:

1441 Budapest
VIII., Reguly Antal u. 8.
Telefon és fax: 133-1839

Felelős kiadó:
Sebestyén Ilona
ügyvezető igazgató



Cédrus Kiadó Kft

Nyomdai előkészítés:
Tipoprint Kft, Budapest

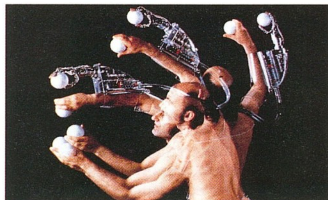
Nyomtatás:
Zalai Nyomda, Zalaegerszeg
Felelős vezető: Galla József

Terjeszti a Magyar Posta.
Előfizethető a hírlapkézbesítő
postahivataloknál és a Posta
Hírlapelőfizetési és Lapellátási
Irodájánál (XIII., Lehel u. 10/a,
Budapest 1900), vagy átutalással
a 215-96162 pénzforgalmi számra.

Példányonkénti ár: 196 Ft
Évi előfizetési díj: 2 352 Ft

Külföldre terjeszti a Kultúra,
Pf. 149, Budapest 1389

HU ISSN 0865-9788



Az egész szakma szerszáma

„Magzarul besyelunk" — ezt a címet adtuk a hónap kiemelt témájának több mint két éve, amikor útjára indítottuk az Alaplapot, s azóta is többször visszanyúltunk az akkor felvetett kérdéshez: mennyire képes a számítógép arra, hogy a magyar nyelvű szövegeket korrektül feldolgozza. S noha ezen a lerághatatlan csonton most is sokat tudnánk még szortyogni, ezúttal a dolog másik aspektusára keressük a választ: mennyire ügyes szerszám mindennapos munkaeszközünk, a magyar számítástechnikai szaknyelv.

Számunkra ez a téma a szokásosnál szubjektívebben vetődik fel, mert egész lapkészítési munkánk alapvető koncepcióját érinti. Szó sincs persze arról, hogy ne adnánk ebben is ugyanúgy teret minden ellentétes véleménynek, de a felvetett kérdésekben az Alaplap a számítástechnikai ismeretek terjesztéséhez alkalmazott stílusával, szóhasználatával, írásmódjával eleve kifejez bizonyos nyelvi koncepciót, tudatos törekvést. Felfogásunkat az olvasó a lap egészéből is leszűrheti, de az egyértelműség — és az esetleges vita — kedvéért ebben a bevezetőben is megfogalmazzuk.

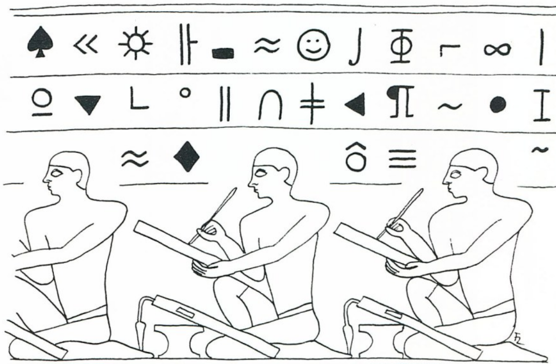
Nyelvi eszköztárunk kialakításához mindenekelőtt végig kellett gondolnunk, hogy várhatóan milyen formában honosodik meg a számítástechnika a magyar nyelvben.

1) Elszigetelt szakmai tolvajnyelv lesz, amelynek nincs is szüksége a köznyelv irányába építendő hidakra? (Lásd vegyészet.)

2) Misztifikált tudományos tolvajnyelv köntösét ölti magára, amely mellett párhuzamosan létezik azonban egy hétköznapi szóhasználat is? (Lásd orvostudomány.)

3) Közérthető szakmai kifejezésekkel dolgozik, s így könnyebben érteti meg széles körben önmagát? (Lásd történettudomány.)

Az első változat inkább csak olyankor életképes, ha egy adott szakterület megmaradhat kevesek mesterségének (harangöntés), eredményeinek élvezetéhez (harangszó) nem szükséges ismerni a gyártási technológiát, de még a késztermék működtetésének módszereivel is csak egy igen szűk rétegnek kell tisztában lennie (harangozó). Talán nem szükséges bővebben fejtegetnünk, hogy a számítástechnika számára ez az út miért nem lenne járható.



A második lehetőséggel a számítástechnikai szakemberek is sokáig kacérkodtak. A PC megjelenése előtt a nagygépes adatfeldolgozási bérmunkát ugyanolyan mesterséges nyelvi köd lebegte körül, mint az orvosi zárójelentéseket. Ez a „kettős könyvelés" a számítástechnikában sokkal nehezebb lenne, mert a torokgyulladásról még csak van fogalmunk annak latin neve nélkül is, de a lemezen lévő hibás cluster megtalálásához nincs semmilyen velünk született „közvetlen laikus eljárás".

A számítástechnikai eszközök és programok használatához a jelek szerint elengedhetetlen feltétel működési módjuknak — alkalmazási területtől függő szinten való — megértése, s a kettő között a szaknyelv az összekötő kapocs, amit egy eljövendő „fekete doboz" korszak sem tehet feleslegessé.

Szinte kizárásos alapon jutottunk el tehát oda, hogy a számítástechnika szaknyelvének szükségszerű útja a közérthetőség. Ebben nagyon hasonlít például a gazdasági terminológiához, amelyet szintén az élet minden területén kell érteni és használni, mégpedig egységesen, egyértelműen és szabatosan, akár egy nagyvállalati főkönyvelő ír mérlegbeszámolót, akár egy magánkereskedő nyújt be hitelkérelmet.

Miután tehát megítélésünk szerint a majdan kiforrottá váló magyar számítástechnikai szaknyelvre nem az elkülönülés, hanem a köznyelvhez való szerves illeszkedés lesz jellemző, mi kezdettől fogva igyekeztünk még a számítástechnikai boncnyelven megírt, agyonkódolt szakanyagokból is minél több ember számára érthető, élvezetes írásokat szerkeszteni.

Azzal is tisztában voltunk viszont, hogy a szaknyelvnek is megvan a maga természetes fejlődési folyamata, amit nem lehet erőszakosan és radikálisan megváltoztatni, legfeljebb csak finom lépésekben befolyásolni és korrigálni. Ezért egyaránt elzárkóztunk az erőltetett, mindenáron való magyarítási törekvésektől és az angol terminológiát tűzön-vízen át előnyben részesítő xenomániától.

Az én nyelvem, a te nyelved, az ő nyelve

A tűzcsapból is víz folyik

A szociológusok szerint a szakember abban különbözik a hétköznapi embertől, hogy időnként olyasmit mond, amit a kívülállók nem értenek meg.
Mi, számítógépesek, jó szakemberek vagyunk, és néha túlteljesítjük a normát:
ilyenkor olyanokat mondunk, hogy magunk sem értjük.

Mitől lesz érthetetlen egy mondat? Ennek alapjában véve két oka lehet: vagy ismeretlen szavakat használunk, vagy nem úgy rakjuk össze mondattá a szavakat, ahogy azt a magyar nyelv diktálná. A két jelenségnek egészen más súlya van. Az első, hogy a magyar mondatba idegen szavakat keverünk, egészen mindennapos. Az, hogy a mondat szerkezete nem eléggé magyar, már ritkábban fordul elő, akkor is inkább a fordításokban.

A szótár

A szaknyelv egyik jellegzetessége, hogy szüntelenül új fogalmak tűnnek fel benne. Hogyan hívjuk őket? Az angolok nagy előszeretettel használnak olyan rövidítéseket, amelyek önmagukban is értelmes, sőt az adott fogalmat bizonyos mértékig jól kifejező szavak. Amikor ez nem megy, akkor valami mást találnak ki, de sokkal kevesebbet morfondíroznak rajta. A bit, a chip, a BASIC rendkívül találó szavak. Hogy a stack (rakás/halom) miért az egyik fajta memóriakiosztás neve, míg a heap (halom/rakás) a másik fajtáé, az nyilván csak véletlen műve, hiszen az angol értelmező szótár is az egyikkel magyarázza a másikat és a másikkal az egyiket, számítástechnikailag pedig két elég különböző dologról van szó.

A fogalommal történt megismerkedés után az első lépes nyilván az, hogy úgy használjuk, ahogy leírva látjuk. Itt már kezdődnek a problémák: vagy tudjuk ugyanis a szó kiejtését, vagy magunk „generáljuk", elképzeljük valahogy. Kérdés az is, hogy amit ismerünk, azt angolul tudjuk vagy amerikaiul. Továbbá hogy milyen végződéseket diktál hozzá a fülünk. Ezen nemigen hiszem, hogy lehetne segíteni. Ami az egyik embernek 'standard', az a másiknak 'sztenderd', a scheduler az egyik embernek 'sedjuler', a másiknak 'szkedjuler' — aszerint, hogy korábban hol találkozott a szóval.

Ezután több dolog történhet. Lehet, hogy az idegen szó lassan magyar szóvá válik, valamilyen formában rögzül a helyesírása, a kiejtése, a ragozása, és végül az Akadémia is áldását adja rá. Másik megoldás, hogy a mechanikusan lefordított idegen kifejezést használjuk, de az is előfordul, hogy valaki talál a fogalomra egy jó magyar szót, és az terjed el.

Gyakoriságát tekintve valószínűleg az első út a tipikus, bár minél távolabb van a szó eredeti alakja a kiejtéstől és a magyaros hangzástól, annál nehezebben megy a dolog. A 'mini'-ből pillanatok alatt magyar szó lett. A makróhoz, lézerhez, processzorhoz sem kellett sok idő. Nehezebben ment mindez a hardverrel, szoftverrel, szervizzel, a fájl szó pedig még ma is éles vitákat vált ki.

Én úgy látom, hogy a sikeres magyarításnak két feltétele van: az egyik, hogy idejében kell megszületnie (mielőtt az idegen szó túlságosan elterjedne), a másik pedig az, hogy az eredeti angol szó ne legyen túlságosan találó, viszont a magyar szó legyen az. Erre a legjobb példa az 'értékadás', ami tényleg sokkal jobban megragadja a dolog lényegét, mint az angol assignement (hozzárendelés, kijelölés). A magyar verem szó is sokkal jobb egy olyan tárolóra, amelynek csak a tetejéről lehet elvenni, és oda lehet tenni, mint a stack (azaz kazal), amiből bizony ki lehet huzigálni egy kis szalmát oldalról is.

Távol álljon tőlem a nyelvvédők bírálata, de az elmúlt 35 évben megéltem egy tucat magyarítási kampányt, és azokból nem emlékszem egyetlen szóra sem, amely végül is kiszorította volna a kiszorítandó szót.

A fordítás szerintem akkor válik be, amikor az eredeti szónak nincs semmi rejtett jelentése, és a lefordított szó is ugyanazt jelenti. Világos például, hogy a 'line printer'-t sornyomtatónak kell nevezni. Persze a korrekt fordítás igen kemény dió. Az elmúlt hónapban két kolléga is hosszú előadást tartott nekem arról, hogy mekkora marhaság a RAM-ot (random access memory) véletlen hozzáférésű tárnak nevezni. Egyesek szerint jobb lenne, ha tetszőlegesen vagy találomra elérhető tárnak hívnánk. Hát én ezt sem érzem sokkal jobbnak, hiszen valójában arról van szó, hogy a tár bármelyik egysége közvetlenül, a cím alapján elérhető, így tetszőlegesen, ha akarom véletlenszerűen helyezhetők el a tárban az adatok, és ez nem befolyásolja elérési idejüket. Mindez bizony az eredeti angol kifejezésből sem igen tűnik ki, miért lenne hát a tükörfordítás sokkal jobb. De végül is ez nem tragédia, az elnevezés csak elnevezés, a



— ...az új anszisziszsszel sikerült végre bútolni, és rögtön debaggoltam a baszfájlt...

tűzcsapból is meg a vízcsapból is víz szokott folyni.

Vannak azután felemás megoldások, mint az operációs rendszer. Vajon miért nem hívjuk kezelő vagy működtető rendszernek? Hiszen az a funkciója. Amennyire vissza tudok emlékezni a dologra, ez az alternatíva már a kezdet kezdetén felmerült, de a többség nem találta elég szakszerűnek. Azt mondták, hogy az orvos is latinul szokta megnevezni az ember szerveit, pedig a legtöbbnek van becsületes magyar neve.

Sajátosan magyar — és elsősorban nyelvészek által javasolt — megoldás a visszalatinosítás. Vagyis ha az angol szó eleve latin eredetű, térjünk át az angol kiejtésről a magyar hagyományok szerinti latinos kiejtéshez, szóalakhoz. Az egyik ilyen kampány azért indult, hogy ne 'kompjútert' mondjunk, hanem 'komputert'. Annyit értek csak el, hogy sikerült a kiejtés sokféleségét megnövelni. Mostanában többen elmagyarázták nekem, hogy 'projekt' vagy 'prodzsekt' helyett helyesebb volna a 'projektum' használata, hisz ez az eredeti latin szó. Valószínűleg ezzel is csak az előforduló variációk számát növelnénk.

A nyelvtan

Az emberek általában úgy vélik, hogy a szaknyelv sokkal pontosabb, egyértelműbb, mint például az irodalom nyelve. Korábban voltak olyan feltételezések, hogy a szaknyelv egzaktságának forrása a korlátozottabb — vagy legalábbis a hétköznapi beszédétől és az irodalométól arányaiban eltérő — nyelvtan használata. Noha pontos statisztikát nem csináltunk, elég sok különböző szöveget elemeztünk géppel, és nem tűnt fel, hogy valamilyen nyelvtani szerkezet hiányozna vagy éppen a szokásosnál sokkal gyakrabban fordulna elő a számítógépes szövegekben. Apreszján professzor, aki rengeteg elektronikai szöveget géppel fordított angolból és franciából oroszra, szintén azt mondta, hogy ha egy nyelvtani konstrukció létezik, akkor azt használják, bármilyen témáról legyen is szó.

Ezzel kapcsolatban szeretném felhívni a figyelmet két nyelvi babonára. Sokan tudják úgy, hogy a szenvedő mondatszerkezet magyartalan, illetve germanizmus. Ez így nem igaz! Ha azt mondom, hogy: „bele van gyűrődve a nyomtatóba a papír", az ugyanolyan jó, mint az, hogy „zöldre van a rácsos kapu festve" vagy „be van fejezve a nagy mű, igen". Mindazonáltal ez a szerkezet mégis viszonylag ritkán szerepel az alany állapotának jellemzésére. És itt jön a másik babona. Egyesek azt hiszik, hogy magyarosabb lesz a mondat, ha névszói szerkezetűvé alakítják, amiben egy múlt idejű melléknévi igenév az állítmány. Például ahelyett, hogy „az A és B pont a piros vezetékkel van összekötve", azt mondják, hogy az „az A és B pont piros vezetékkel összekötött". Hát ez egy óriási marhaság, mert a múlt idejű melléknévi igenévi állítmány igen kivételesen előforduló forma. Számos nyelvész szerint ilyen mondat nincs is, mert múlt idejű melléknévi igenév nem lehet állítmány, csak ha igei jellegét elveszítve már melléknévvé vált.


—Ha így ejtem ki a kompjútert, a nyelvészek talán nem kötnek bele.

A stílus

A egzaktság mellett a szaknyelvről azt is feltételezik, hogy meglehetősen száraz. A költemények tele vannak érdekes hasonlatokkal, allegóriákkal, megszemélyesítésekkel. A prózai művekben is vannak ilyenek, de elég ritkán fordulnak elő a számítástechnika nyelvében. Mi is meglepődtünk, amikor egy szigorúan technikai szövegben először találtunk megszemélyesítést. Ha jól emlékszem, egy lemez „haldoklásáról" volt szó, annak kifejezésére, hogy valamely hiba miatt a lemezen egyre több szektor vált használhatatlanná.

Vannak persze szövegek, például a referencia-kézikönyveké, amelyeket azért veszünk kézbe, hogy megnézzük: ezt ide kösd, amazt így állítsd be, emennek ilyen paraméterei vannak. Ilyen esetekben engem is zavarna minden „lazaság". Legyen az ilyen szöveg teljesen pontos és tömör. Más a helyzet akkor, amikor valamit el kell magyarázni. Azt szokták mondani, hogy egy jó ábra többet ér, mint két oldal szöveg. De a szakemberek közül kevesen tudják, hogy egy szókép is sokat segíthet valaminek a megértetésében.

Egyes sikeres szerzők részletesen kifejtett allegóriákat alkalmaznak egy bonyolultabb téma ismertetésekor. Különösen szép példákat találunk a multitasking rendszerek leírásánál. A 'task'-ok születnek és meghalnak, gyermekeket szülnek vagy osztódnak, randevúznak egymással, versengenek az erőforrásokért, amelyik pedig nem jut hozzá, azt kiéheztetik vagy kilövik.

Persze felmerül a kérdés, hogy vajon a „kiéheztetés" hasonlat-e vagy szakkifejezés. Biztos vagyok benne, hogy először az „intelligens" szót is szimbolikus jelzőnek szánták, ma pedig az „intelligens terminál" meglehetősen pontosan, majdhogynem formálisan definiálható. A „barátságos" (felhasználóbarát) szó talán még nem jutott el erre a fokra, de azért már többé-kevésbé pontosan tudjuk, hogy mit jelent.

Összegzés

Mint minden nyelv, a magyar is igen sokféle. Más az én nyelvem, a te nyelved, az ő nyelve, ha én számítástechnikus vagyok, te kórboncnok, ő meg alkotmányjogász. Persze egy negyedik témában (különösen, ha egyikünk sem ért hozzá igazán) igen jól el tudunk beszélgetni. És más volt a nyelv tegnap, és más lesz holnap. Még inkább igaz ez a számítástechnika nyelvére, amely a szakmával együtt rendkívül dinamikusan fejlődik. De azért valahol a mélyben minden szaknyelv a hétköznapi magyar nyelvvel azonos szabályok szerint építkezik. Az nem várható el, hogy valaki egy komoly szakcikket minden előzetes ismeret nélkül megértsen, már a nyelv szintjén sem, hiszen számára idegen szavakkal és friss magyarításokkal kellene megküzdenie. De annak, ami ma szaktudás, egy része holnap közkincs lesz. Ezért mindig keresni kell az új fogalmak szabatos és egyértelmű magyar megnevezését, óvatosan kell bánni a tükörfordításokkal, és törekedni kell a szemléletességre.

Farkas Ernő

6:3 helyett döntetlen

Angol–magyar mérkőzés

Az utolsó egy-két évtizedben a tudományos-műszaki életben talán a számítástechnika fejlődése volt a legsikeresebb és a leglátványosabb. Az új tudomány és annak gyakorlati alkalmazása elkerülhetetlenül felvetette a vele kapcsolatos új nyelvi kérdéseket is.

A 80-as évek közepén a magyar sajtóban csaknem fél évig vitatkoztak arról, hogy milyen, illetve milyen legyen a számítógépes szaknyelv. Csak néhány kiragadott idézet a vélemények közül:

„Itt az ideje, hogy ne rontsuk tovább nyelvünket... már régóta forr bennem a nyelvérzéki indulat két, írva, nyomtatva, olvasva, kimondva képtelen szó ellen... Természetesen a *hardver* és a *szoftver* szószörnyszülöttekre gondolok. Míg az eredeti angol *hardware* és *software* alakjukban megjárta, legalább olvasva nem bosszantotta az embert. De fonetikusra átmagyarítva elviselhetetlen is, értelmetlen is." (Boldizsár Iván: Kemény és lágy. Népszabadság, 1985. XII. 24.)

„Az egységes számítástechnikai szaknyelv... alapja Magyarországon is csak az angol lehet... *Ne ott magyarítsunk*, ahol az erőszakolt magyar szóhasználat az egyes szakterületeken való elmaradottságunk megőrződéséhez vagy fokozódásához is hozzájárulhat. Számoljunk le végre azzal az illúzióval..., hogy honi magyar nyelvünk valaha is alkalmas lesz a világméretű haladás naprakész követésére..." (Oláh Zoltán: Számítástechnika és leányvállalat. Magyar Nemzet, 1985. XII. 29.)

„A nemzetközi hordaléknyelv divatszavainak, a kereskedelem, a vendéglátóipar és a számítástechnika főként angol szakkifejezéseinek beözönlése elleni harc nálunk többindítékú akadályokba ütközik... Vannak, akik egyenesen nyelvünk gazdagodásának minősítik a minél több internacionalista kifejezés meghonosítását. Ez az álláspont a legveszélyesebb, mert a magyarságtudat elhalványodásához, a nemzeti nihilizmus terjesztéséhez vezet." (Antalffy Gyula: A számítógép nyelve. Magyar Nemzet, 1986. I. 11.)

„Mindaz, ami a köznyelvben kérhet jogot arra, hogy a számítástechnikai használattal *analóg* kifejezésként szerepeljen: legyen magyar. Ugyanakkor nem tiltható meg a szaktudománynak, hogy kihasználja a számítástechnikai szaknyelv nemzetközi előnyeit..., mert nem a magyart rontja, s mert nem az angol honi szálláscsinálója." (Ungvári Tamás: Eretnek nézetek a szakmai nyelvről. Magyar Nemzet, 1986. II. 18.)

„Az *angolos* terminológia melletti legerősebbnek tűnő érv az, hogy az angol biztosítja a szaknyelv nemzetköziségét. A nemzetközi terminológia azonban mítosz... Nem elég a rossz hangzású szó miatti felháborodás és a javító szándék. Szakmailag meg kell tudni ítélni, hogy a létrehozott szó alkalmas-e az adott fogalom kifejezésére, és van-e lehetőség az elterjesztésére. Ehhez viszont szakértelem kell, nem elég a jó szándék." (Kiss Ádám: Szoftver és türelem. Magyar Nemzet, 1986. III. 17.)

„Feltűnően sokat beszélünk a számítástechnika nyelvéről; hajlok arra, hogy sok esetben azért borzongunk a nyelv szörnyűségétől, mert valójában az új technika birtokbavételétől ódzkodunk." (Mérő László: A tenni kéne valamit szindróma. Magyar Nemzet, 1986. III. 24.)

A számítástechnika nyelvezetéből írt cikkeknek természetesen csak elenyésző hányadából idéztem, de talán ennyi is elegendő ahhoz, hogy nyilvánvalóvá váljék: az egyes vélemények között kibékíthetetlen ellentétek vannak.

A szakemberek jelentős része úgy véli, hogy mindenképpen ragaszkodnunk kell az eredeti angol nyelvű terminológiához, s ennek megfelelően át kell vennünk és használnunk az idegen nyelvű szakkifejezéseket. Érveik közül a leggyakoribb a szaknyelv nemzetközisége, s írásaikból az hallik ki, hogy az új technika befogadásához a nyelvi elemek átvétele is szükséges, különben szakmai elmaradottságunk is növekedhet.

Mások viszont élesen kikelnek „a nemzetközi hordaléknyelv divatszavai" és az „angol kifejezések tébolyító dömpingje" ellen.

A dolog szépséghibája mindössze az, hogy sokszor olyan emberek is elítélően nyilatkoznak a számítástechnikai szaknyelvről, akik a *hardver* és a *szoftver* szón kívül egyáltalán nem ismerik a számítástechnikai szakszavakat, számítástechnikai ismeretekkel pedig egyáltalán nem rendelkeznek. Az önmagában helyes, hogy lehetőleg magyar szavakat használjunk, de a szakterület valamelyes mértékű ismerete nélkül nyilatkozni a szaknyelvről meglehetősen vitatható dolog.

Kissé sarkítva: a hozzászólók egyik csoportja szerint a számítástechnika szaknyelve csak az angol lehet, hiszen illúzió, hogy a magyar nyelv alkalmas lesz e tudomány művelésére. Ellenzékük viszont követeli, hogy e szaknyelv is legyen magyar, mert a nemzetközi szaknyelv, a nemzetközileg egységes terminológia nem több, mint puszta mítosz.

Valójában az eltérő vélemények sokkal árnyaltabbak. A véleményüket nyilvánítók minden esetben más-más határt húznának meg az eredeti angol kifejezések megtartása vagy magyarosítása kérdésében. Érdekes viszont, hogy az egymástól homlokegyenest eltérő vélemények egy kérdésben többnyire azonosak: a többség ellenzi az eredeti angol szakkfifejezések fonetikus átírását, bár rendszerint eltérő okokból.

A fonetikus átírás kompromisszumként született. Az eredeti angol szavak leírása és helyesírási szabályaink szerinti toldalékolása, azaz a magyar mondatokba való beillesztése ugyanis számos nehézséget okozott (*hardware-rel, byte-ot, file-lal* stb.). Akadt kezembe olyan számítástechnikai szakcikk is, amelyben egy bekezdésen belül egy angol szónak ugyanazon magyar toldalékkal ellátott alakja három formában is szerepelt: *file-vel, file-lal, file-lel!*

A magyarítást ellenzők egy része azért tiltakozik a fonetikusan átírt formák ellen, mert szerintük elveszítjük az eredeti angol nyelvi forma használatából fakadó előnyöket, s különben sem a magyar köznyelvben, hanem csak a számítástechnika tudományos nyelvében használjuk őket.

A magyarítást kívánók többsége szerint viszont éppen az a baj, hogy jóllehet elvesztettük az eredeti angol formák ismeretét, mégsem magyarosítottuk meg a szóba jövő kifejezéseket, hiszen a *hardver* vagy a *szoftver* — és sorolhatnám még a példákat — a bennük rejlő mássalhangzó-torlódás és a magánhangzók diszharmóniája miatt, pusztán az írásképet megváltoztatva, semmivel sem lett magyarabb, tehát semmivel sem kerültünk közelebb a problémák megoldásához.

Mindkét táborban voltak és vannak olyanok, akik elfogadják ezt a — mondjuk ki nyugodtan — félmegoldást. Ezt bizonyítja az MSZ 7788 jelzetű, „Az adatfeldolgozás fogalmai" című, a számítástechnika különböző részterületeinek terminológiáját szabályozó szabványsorozat, amelyben szép számmal találhatók fonetikusan átírt szóalakok.

Véleményem szerint az egyes szavak fonetikus átírása nemcsak nem oldja a nehézségeket, az ellentéteket, hanem tovább bonyolítja a helyzetet. A szakírónak, a szaknyelv használójának nemcsak az eredeti angol írásképet kell ismernie, hanem azt is fejben kell tartania, hogy az adott szakkifejezést csak angolosan, csak magyarosan, vagy esetenként mindkét formájában leírhatja-e.

A fonetikus átírás ilyen veszélyeket hordozó mivolta más szakterületen már bebizonyosodott. A Brencsán János által szerkesztett *Új orvosi szótár*-ban látunk erre jó — azaz rossz — példát: 63 *pseudo* előtagú címszó közül 36 esetében utalja az olvasót a *pszeudo* előtagú címszóhoz. Ez azt jelenti, hogy 63 eredetileg azonos előtagú szóösszetétel közül 27 szakszóban — a pseudo, 36-ban pedig a *pszeudo* írásformát kívánja meg. Kötve hiszem, hogy bárki is fejben tartaná, melyik összetett szó előtagjában melyik írásképet tartja helyesnek a szótár. S kötve hiszem, hogy a számítástechnikai szaknyelv amúgy is meglehetősen sok tisztázatlan nyelvhasználati kérdését hasonló írásképi szavakkal kell még tovább bonyolítanunk.

A számítástechnika szaknyelve — más szakterületek nyelvétől eltérően — sajátos kettősségeket hordoz. Maga a szaknyelv is kettős: a programnyelvek (gépi nyelvek) angol eredetűek, az angol írásformákat őrzik, míg a tágabb szaknyelv is angol eredetű, de számos magyar elem is található benne. A programnyelvek magyarításáról érdekes módon nem esik szó. Ehhez ugyanis nem nyelvi, hanem számítástechnikai fejlesztés, azaz önálló magyar számítógépgyártó ipar kialakítására lenne szükség. Eltekintve a gazdasági kérdésektől, erre csak akkor lenne érdemes vállalkozni, ha a magyar termékek színvonala jelentősen felülmúlná a jelenlegi világszínvonalat. Erről pedig jelenleg ábrándozni sem érdemes.

A másik kettősség, hogy az angol szavakat a magyar közleményekben magyar szövegekbe, mondatokba helyezzük bele, s ennek megfelelően magyar toldalékokkal (ragokkal, jelekkel, sőt: képzőkkel is) látjuk el őket.

Nem az a baj igazán, hogy idegen szavakat veszünk át, hanem az, hogy ezek nem illeszthetők jól be a magyar kiejtési és grammatikai rendszerbe.

További ellentmondás, hogy egy-egy angol szót (*cursor*) a magyarok eléggé sokféleképpen ejtenek (kurzor, körzör stb.), sőt időnként az írásformával is gondok vannak. (Pl.: a szóvégi néma magánhangzóhoz kötőjellel kapcsoljuk a magyar toldalékokat: *hardware-es*!)

A harmadik, hogy a számítástechnikai szaknyelvet használók köre két részre oszlik. Az egyik, a számítástechnikai szakemberek, akik otthon vannak saját tudományukban, s ennek megfelelően kimerítően kell ismerniük szakterületük minden nyelvi jelenségét. Legyünk őszinték, s mondjuk ki, hogy megfelelő angol nyelvi ismeretek nélkül igazán ez senkinek sem sikerülhet.

A másik kör a felhasználók köre. Hovatovább a számítástechnika minden tudományba betör, s az összes tudományok összes művelője, ha érdemben hozzá akar járulni szaktudománya fejlődéséhez, kutatómunkájában nem nélkülözheti a számítógép segítségét.

De nemcsak a tudományban, hanem a mindennapi gyakorlati munkaterületeken is jelentős segítséget, könnyítést nyújt a számítógép használata a kereskedelemtől a gyógyításig, az oktatástól a bűnüldözésig.

Ezeket a felhasználókat nem igazán érdekli vagy izgatja, hogy a számítástechnikai szaknyelv valójában milyen. Számukra nem érdekesek az erről folytatott viták sem. Nekik arra van szükségük, hogy kutató- vagy gyakorlati munkájukban hathatós segítséget kapjanak a számítástechnikától. Azaz olyan programok álljanak rendelkezésükre, amelyek megkönnyítik munkájukat. Ha ehhez kevesebb számítástechnikai szakismeretre van szükségük, szívesebben és könnyebben használják a számítógépet. Ha a munkájukhoz szükséges számítástechnkai ismeretek megszerzését bonyolult nyelvi ismeretek elsajátítása vagy egyszerűen a terminológiai zavarok megléte nehezíti, akkor egyértelmű, hogy bizonyos ellenállás mutatkozhat a számítástechnika használatba vételével szemben.

Más mércével kell tehát mérnünk, attól függően, hogy melyik körnek szól a közleményünk. A számítástechnikai szakemberek körén belül szinte mindegy, hogy milyen nyelven folyik egy vita, csak a kölcsönös megérthetőség és az egyértelműség követelményének kell minden esetben eleget tenni.

Ezen túl az, hogy egy-egy műhelyben, pl. egy szaktanszéken a terminológiai választékból melyik kifejezést használják, az az ott dolgozóknak szinte magánügye. Viszont egy publikáció esetén az érthetőség miatt már valamilyen szaknyelvi normához kellene tartanunk magunkat, s ez nem mindig sikerül.

Közbevetőleg jegyzem meg, hogy a nyelvi norma megsértése nemcsak a terminológia vagy a helyesírás területén zavaró. Talán itt a legkevésbé, hiszen még nem is beszélhetünk egyértelműen kodifikált számítástechnikai terminológiai és helyesírási normáról. Számomra az még inkább zavaró, hogy sok szakközleményben az általános magyar nyelvi normákat sem tartják szem előtt, sok a fogalmazási, nyelvi-nyelvtani, szóhasználati pontatlanság, sőt időnként a helyesírási hiba is.

Mi tehát a megoldás? Az ideális a svéd példa követése lenne. Jól felkészült szakemberek (nyelvészek is!) foglalkoznak egy kutatóközpontban azzal, hogy az újonnan megjelent, idegen nyelvű szakszavak, szakkifejezések helyett megfelelő, a nyelvi követelményeket kielégítő svéd nyelvi elemeket készítsenek és segítséget nyújtsanak azok elterjesztésében, illetőleg a normának nem megfelelő nyelvi alakzatok kiszűrésében.

Ez az út nálunk egyelőre nem járható. Kár, mert *minden* szaknyelv fejlődésében nagy segítséget nyújthatna egy hasonló magyarországi intézmény felállítása. Ennek azonban nemcsak személyi, hanem anyagi feltételeit sem tudjuk megteremteni a közeljövőben.

A másik út, amelyen eddig is jártunk — több-kevesebb sikerrel —, az, hogy maguk a szakterületet művelő szakemberek több figyelmet fordítanak a szaknyelv normatív és terminológiai kérdéseire. Ha ez pusztán annyit jelent, hogy

az egyes szakközlemények megfogalmazásakor igyekeznek gondosabban ügyelni a nyelvi megformálásra, már jelentős lehet az előrelépés.

Alapvető kérdés, hogy angol vagy magyar legyen a terminológia. Nem tartom túl szerencsésnek ezt a merev kettévágást. Igenis átvehetjük és megtarthatjuk az eredeti angol formát, ha az nem okoz különösebb kiejtési vagy írásbeli nehézségeket (*bit*).

Próbálhatunk magyar megfelelőt alkotni abban az esetben, ha az angol íráskép és kiejtés a magyar anyanyelvű számára teljesen idegen, és a szó magyar használata bizonyos zavarokkal járhat (*file-lal, fájllal!*) Ez azonban nem sikerült és nem is fog sikerülni minden esetben. Hiszen egy bizottság, amely szakemberekből és nyelvészekből állt, több napon keresztül vitatkozott azon, hogy a *hardware* és *software* angol szavak helyett mit ajánljon. Végül a fonetikus átírásnál jobbat nem tudott javasolni, pedig több tucat javaslatot kapott ülései során. S ma e két szó szinte a mindennapi szóhasználat szerves részévé vált, sőt képzett alakjai (hardveres, szoftveres) is szerepelnek már szókincsünkben.

A számítástechnikai szaknyelv jelenlegi állapota még a legszigorúbb purista szempontból nézve sem rosszabb, mint sok más szakterület nyelvéé. Jóllehet mindent el kell követnünk, már csak a színvonalas szakmai kommunikáció érdekében is, a szaknyelv fejlesztése, egységesítése érdekében, de helyzetét nem érzem tragikusnak vagy rossznak.

Figyelembe kell venni azt is, hogy szerte a világon a számítástechnika, illetve annak alkalmazása fejlődik talán minden szakterület közül a leggyorsabban. Ez nyelvi szempontból azt is jelentheti, hogy bizonyos, sokak által kifogásolt szavak vagy kifejezések éppúgy a lomtárba kerülnek, mint azok az eszközök vagy eljárások, amelyek jelentését hordozzák.

Arra is számíthatunk, hogy a fejlődés mindig új nyelvi kérdéseket is felvet, vagyis állandó az új angol szakkifejezések és szakszavak utánpótlása. Ha egy új fejlesztés, egy új technikai elem vagy eljárás átvételekor figyelembe vesszük — vagy legalábbis megpróbáljuk figyelembe venni — a szaknyelv, tágabb értelemben véve a magyar nyelv *fejlődésének* a követelményeit, nő az esély arra, hogy belátható időn belül egy jobb, tisztább, esetleg egyszerűbb, magyarosabb, a felhasználók számára is könnyebben érthető szaknyelvet alakíthatunk ki.

Seregy Lajos

A „beépített ember"

Olvass.el (!)

Ugye, milyen sokszor találkozunk
a cikk címéül választott fájlnévvel?
És ugye, idétlen a külön zárójelbe tett felkiáltójel a végén?
Ha ugyanis fel akarom szólítani valamire az olvasót,
akkor miért tompítom mondandómat egy zárójellel,
ha pedig csak mellékesen,
zárójelben akarok valamit megjegyezni,
akkor miért egy kiugrató jelet teszek bele.
Ez a példa egy a sok közül
a szerkesztés során keletkező nyelvi ellentmondásokra,
amelyek feloldása igen gyakran
az olvasószerkesztő feladata.
Az alábbi eszmefuttatás az ő nézőpontjából igyekszik
bepillantást adni a szaknyelv gyúrásának küzdelmeibe.

Az olvasószerkesztőt segédszerkesztőnek is szokták nevezni, ami jól utal kettős helyzetére. Egyrészt ő a főszerkesztő jobbkeze a kéziratok gondozásában, másrészt ő az olvasó „beépített embere" a szerkesztőségben. Az egyik gazdasági lap főszerkesztője soha nem korholta a szabadkozó fiatal olvasószerkesztőt, ha az nem értett valamit egy cikkben, hanem inkább dicsérte: „Fiam, ez a dolgod, hogy jelezd! Ha te nem érted, akkor nagy a valószínűsége, hogy az olvasó sem fogja érteni, tehát korrigálni kell a fogalmazást." Ha csupán nyelvi ügyetlenség zavarja a megértést, azt a rutinosabb olvasószerkesztő maga is ki tudja javítani, de ha szakmai értelmetlenségek, logikátlanságok, ellentmondások vannak mögötte, akkor bizony konzultálnia kell a szerzővel.

Szabad a pálya

A fentieket nem azért írtuk le, hogy népszerűsítsünk egy olyan pályát, amely felé nem nagy a tülekedés. (Bár az sem ártana, hiszen az Alaplapnál is betöltetlen az olvasószerkesztői státusz, és a szerkesztőknek kell elvégezniük a végső csiszolás műveletét, egyik laptársunk kiadója pedig immár csaknem fél éve kínálgatja hiába az olvasószerkesztői állást a Magyar Sajtó c. lap hasábjain, miközben újságírók százai vannak munka nélkül!) Inkább arra igyekszünk felhívni a figyelmet, hogy a számítástechnika elterjedése lényegesen megváltoztatta az írásos dokumentumok, műszaki leírások, kiadványok készítésének menetét. Tömegével kerülnek forgalomba úgy az anyagok, hogy nem is látja azokat hivatásos szerkesztő, olvasószerkesztő, lektor vagy korrektor, ezáltal a legértékesebb tartalmat is agyonütheti a nyelvtani, stiláris és helyesírási hibák tömkelege. (A bajt még tetézheti a „lézerprinteresített" tipográfiai analfabetizmus... de az már egy másik mise.) A helyzet megváltoztatására sok-sok „amatőr" olvasószerkesztőnek kellene működnie mindenütt, ahol kéziratokból bármilyen dokumentum érlelődik. Közülük azután néhányan talán ebben lelnék meg igazi hivatásukat is.

Egy számítástechnikai szaklap olvasószerkesztőjétől joggal várható el, hogy tudjon magyarul — és természetesen angolul, németül, latinul, görögül... no meg basicül, prológul, pascalul, modulául, adául, lipsül, fortranul, assemblyül, péelegyül, clipperül, épluszpluszul... De azért mindenekelőtt magyarul. Hasznos, ha volt hozzá egy kis előiskolája is, mert szüksége lesz a helyesírásra, a stilisztikára, a nyelvhelyességre, a szintaxisra, a szemantikára, az etimológiára... És egy jó adag természetes nyelvérzékre.

Egy nyelven beszélni

Ugorjunk csak vissza az iménti nyelvi felsorolásra, hogy mennyi megoldatlan

magyarosítási, helyesírási és kiejtési probléma sűrűsödhet össze néhány szóban is. Sok vitás kérdésben az olvasószerkesztő persze nem lehet döntőbíró, sőt a kapott tanácsok is ellentmondóak, de a lapnak mindenképpen meg kell jelennie, és valahogy le kell írni azt is, amiben még sem a gyakorlat, sem a nyelvészek elvi álláspontja nem egységes.

Az a legjobb iránytű, ha mindig mindennek az értelmét, a jelentését keressük, mert a nyelv ezen keresztül szinte automatikusan ellenőrzi a tartalmat. Nincs nagyobb öröm egy olvasószerkesztő számára, mint amikor egy olyan szövegben, amelyhez igazán mélyen tulajdonképpen nem ért, pusztán a nyelvi okoskodás alapján felfedezi a szakmai tévedést.

Mindenki közös érdeke, és egyben a távlati cél is mindenképpen az, hogy az egész szakma „egy nyelven" beszéljen. Ez taktikailag valószínűleg úgy közelíthető meg, hogy előbb legalább a nagyobb „műhelyek" (könyv- és lapkiadók, szoftvercégek, oktatási intézmények stb.) igyekezzenek házon belül egységes terminológiát és írásmódot alkalmazni, később pedig sor kerülhet egy „csúcstalálkozóra", és annak alapján a normák folyamatos összehangolására.

A szándék szelleme

Természetesen lehetetlen minden cikket a magunk szájíze szerint teljesen átírni, de lehet azokat bizonyos nyelvi normákhoz úgy igazítani, hogy megőrizzük a szerző stílusát, a mű eredetiségét. Az út viszont, amely idáig vezet, igencsak hosszú és keserves:

— meg kell érteni (de nem megtanulni) azt a szakterületet, amelyikkel a cikk foglalkozik,

— követni kell az akadémiai iránymutatásokat (amíg csak lehet), de nem szabad visszariadni a saját nyelvi normák kialakításától,

— folyton fel kell tennünk kétkedő kérdéseinket, hogy érthető-e mindaz, ami le van írva,

— végül pedig rá kell szánni az időt a bizonytalan kérdések tisztázására, minél több mérvadó vélemény ütköztetésével.

A hazai számítástechnikai lapok egy részénél a hőskorszakra jellemző lelkesedéssel vezettek egy folyamatosan aktualizált „szokványjegyzéket", amiből ki lehetett puskázni a lapba bekerülő új szakkifejezések egységes írásmódját. Ez a „mit-hogyan" gyűjtemény azonban máig sem állt össze igazi számítástechnikai nyelvhelyességi szótárrá, valószínűleg azért, mert egy ilyen vállalkozás a benne részt vevőktől összefogást, a kenyéradó gazdáktól független együttműködést és elég sok többletenergiát igényelt volna — s mindez nem volt meg. Talán csak a szándék szelleme őrködött azon, hogy ennek ellenére valamennyi szaklapnál megmaradt egy olyan szemlélet, amely a lényegi kérdésekben hasonló elveket tükröz.

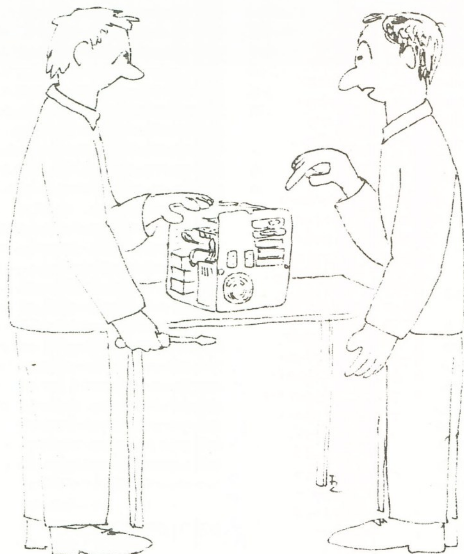
Mindenki szavaz

A szaknyelv magyarításában mindig realistának kell lennünk. Ha egy új szakmai fogalom bukkan fel, és olyan magyar szót találunk rá, amely pontos, kifejező és jól használható, azt a szakma is nyomban elfogadja, s csak a „pufferben" őrzi meg az angol eredetit. Ehhez viszont valóban telitalálatnak kell lennie...

A „mindent mindenáron magyarul" megoldás gonosz csapda, a szakmai nyelvhasználattól túlzottan idegen magyar kifejezések erőltetése éppen fordított hatást ér el: nevetségessé válik, ami kihat még a későbbi — és esetleg már valóban jó — kísérletek fogadtatására is. A szakmai nyelv fejlődése öntörvényű folyamat, s bár valamennyien részesei vagyunk a törvényalkotásnak, a beterjesztett indítványokról az egész „parlament" szavaz. (A szakmát pedig nem lehet leváltani, még akkor sem, ha a döntéssel elégedetlenek vagyunk.)

A hazai számítástechnika hajnalán a „bájtfülűek" körében kialakított bükkfanyelv oka nem annyira a majmolás volt, mint inkább a hiányos tudás. Nem is születhetett meg olyan ütemben e szakterület nyelvi infrastruktúrája, amilyen sebességgel jövevényként behatolt társadalmunkba. Ahogy azonban ilyenkor lenni szokott, igen sok kókler is szakembernek tűnhetett, ha képes volt a divatos szópetárdák durrogtatására. Ma már, a piac kibővülése és nyitottá válása óta a pusztán nyelvi ködösítéssel manipulálók korszaka is lejárt. Egyre több mindent lehet egyszerű hétköznapi nyelven, mégis egyértelműen és pontosan megfogalmazni, s egyre kevesebben hiszik azt, hogy a szakszerűség és tudományosság a számítástechnikában kizárja a közérthetőséget.

Faklen Pál—Varga János



— A ventilátorig még értem, amit mondtál...

A keresés kritikus kritériumai

Szövegekben böngészve

Tapasztalatom szerint a nyelvészek, ha valami problémájuk van, először más nyelvészek írásaihoz fordulnak, azután beszélnek róla kollégáikkal, és végül, de nem utolsósorban (sőt talán elsősorban) céduláznak: éveken keresztül, sokszor egy életen át gyűjtik az idézeteket az egyes nyelvészeti kérdésekkel kapcsolatban.
Én viszont, mint számítógépen nevelődött ember, ilyenkor is rögtön „a komputeremhez kapok".

Mióta a magyar nyelvű interfésszel és a gépi fordítás kérdéseivel foglalkozom, minden gépen, ahol csak dolgozni szoktam, van vagy tíz-húsz megabájtnyi szöveg. Ezeknek azután neki lehet esni szövegszerkesztővel, keresőprogrammal, szöveges adatbázis-kezelővel vagy különböző ad hoc programokkal, és megnézni, hogy mi is van bennük — túl a szövegek tartalmán. A piac már számtalan ilyen kereső-elemző programot kínál a szövegállományokhoz, és az igény is egyre nő irántuk. Legtöbben azonban nyilvánvalóan egészen más célzattal keresgélnek a számítógépen tárolt szövegekben, ezért talán érdekes lehet számukra, hogy én mire figyeltem fel.

A kóddal nincs szerencsénk!

Eddig a magyar szövegek 60-65 százaléka CWI kódban volt, 25-30 százalék magyar Ventura kódban, s csak a maradékon osztozkodtak az elszigetelt egyéb kódkiosztások. Most a Windows rohamos terjedésével és a 852-es tábla megjelenésével újabb problémáknak nézünk elébe. Itt nem akarom részletezni, miért szemét (angolul: garbage) az egész 852-es, aki esetleg még nem tudná, olvassa el az Alaplap 1992. áprilisi számában vagy a magyar Chipben. Az új kód eléggé ehetetlen ahhoz, hogy a magyar gyomor bevegye, de ilyen téren másutt sem megnyugtató a helyzet. A németek is széles körben használják a német ASCII-t, ezáltal a C programokban umlautos betűk jelennek meg a szögletes és kapcsos zárójelek helyén.

Mindenesetre egy ilyen témájú és drága pénzen magyarított programnak igazán lehetne cserélhető kódtáblája, azaz lehessen meghatározni, hogy melyik kódszámhoz milyen karakterkép tartozzon. Az sem árt, ha egy programnak van billentyűzetátdefiniáló programja, hogy a magyar betűket is le tudjuk vele ütni, de még jobb, ha a megszokottat használhatjuk.

„Keresni csak pontosan..."

A visszakeresés alapfeladata bizonyos szavak és szókombinációk megtalálása a szövegben. Tegyük fel, hogy egy hírözönből ki akarjuk válogatni az almatermeléssel kapcsolatosakat. Ha csak annyit írnánk, hogy keresd az „almatermesztés" szót, valószínűleg igen gyatra eredményt kapnánk. Ennek az az oka, hogy a magyar ragozó nyelv. Az angolban egy főnévnek legfeljebb két alakja van (egyes és többes szám), egy magyar szó viszont — amint azt az Alaplap 1992. májusi számában sokoldalúan bemutatták — több ezer alakot ölthet. Még szerencse, hogy a ragok a szó végén vannak, mert így minden szöveges adatbázisban keresgélhetünk jobbról történő csonkolással. Ha tehát azt írjuk, hogy „almatermesztés$", akkor megtaláljuk az „almatermesztéssel", „almatermesztését", „almatermesztési" stb. szavakat is.

Persze így sem találjuk meg az olyan mondatokat, hogy: „Almából az idén kevesebbet termesztettek." Ez a mondat rögtön féltucat problémát vet fel. Az első az, hogy a magyarban számos kapcsolatot lehet egybe- és különírni. Egybeírjuk a jelöletlen birtokos kapcsolatot (lásd a fenti példát), a jelöletlen határozói kapcsolatot, de különírjuk pl. a birtokos és határozós a jelölt kapcsolatot. Az igekötő és ige különírása és egybeírása is ismert. És ez nemcsak az igére vonatkozik, hiszen a „megoldható" egybeírandó, de az „oldható meg" kettőbe. Ilyenkor azt kell előírni, hogy két szót keressünk, s persze nem akárhol. A magyarban természetes egység a mondat, tehát a mondatokat kell felismerni, megjegyezni.

Kaméleon betűk

A szöveges adatbázisok gyakran rendelkeznek olyan lehetőségekkel, hogy a keresett szó állhat a másik előtt vagy után, álljon ennyi meg ennyi szón belül. Ennek a magyarban igen ritkán van értelme, legfeljebb a többszavas neveknél.

A mondatot a legtöbb program úgy tekinti, hogy a szövegnek pont, felkiáltójel vagy kérdőjel által határolt része. Persze egy tizedespont vagy egy rövidítést lezáró pont kissé megzavarhatja a dolgot. Biztosabb az algoritmus, ha azt is ellenőrzi a program, hogy van-e utána szóköz és nagybetű, de még így is akad benne elég sok bizonytalanság. (Például tulajdonnév jön a ponttal lezárt rövidítés után.)

Tehát egy mondaton belüli szópárokat kell keresni, például azt, hogy „alma$" és „termesztés$", de „az almának a termesztését" mégsem fogjuk megtalálni, mert az „alma" nem „almá". Amikor a magyarban egy szó egy raggal találkozik, gyakran az egyik vagy a másik, esetleg mindkettő átalakul. Jellemző változások: magánhangzó hosszabbodása (béke — békét), magánhangzó rövidülése (kéz — kezet), magánhangzó kiesése (borjú — borjak), a szóvég teljes megváltozása (hisz — higgyük, szó — szavak) vagy egyidejűleg több ilyen jelenség (szép — szebb) stb. Vannak továbbá olyan esetek, amikor a kiejtés nem változik, de a betűkép igen (húsz — hússzor). Kérdés, hogy milyen gyakori ez a jelenség, és hány tőváltozatot kell keresni egy szónál. A szavak jelentős részének egyetlen töve van, egy tucat igének hat

lehetséges tövével találkozhatunk, de átlagosan 1000 tőre 2500 tőváltozat jut. Persze ha van benne némi gyakorlatunk, és ha egy kicsit utánagondolunk, meg tudjuk mondani, hogy egy adott esetben milyen tőváltozatokat kell keresnünk, a dolog azonban semmiképpen nem megy automatikusan.

Lépjünk eggyel tovább. Akkor sem fogjuk megtalálni az „almatermesztést", ha „jonatánalma-termesztés"-ről lesz szó. A „jonatánalma" esetében mindkét oldalon történő csonkolásra van szükség, ami kicsit munkaigényesebb, ezért nem minden programban van megoldva.

Helyesírási akadályverseny

Tényként kell elfogadni, hogy a forgalomba kerülő programok túlnyomó hányada angol (pontosabban amerikai angol) vagy német nyelvi közeg számára készült, a magyar nyelv követelményei sok szempontból eltérőek: nálunk bizonyos eszközök hiányozhatnak, mások viszont feleslegesnek tűnnek. Reméljük, hogy néhány hónapon belül elkészül egy magyar nyelvű szöveges lekérdező interfész terve, amit majd kivitelezni is kell, de a programokat addig is hozzá lehet mérni.

Ha azt hisszük, hogy egy kinyomtatott könyv, újság, törvény vagy szabvány helyesírása hibátlan, természetesen tévedünk, de a helyzet ennél is sokkal rosszabb a belső dokumentumok, fordítások, jelentések és hasonló kéziratok esetén. Tulajdonképpen az lenne célszerű, ha visszakeresés előtt minden szöveget végigpásztáznánk egy szóellenőrző programmal, és kijavítanánk. Elég nagy munka, de ha a szöveget sokat forgatjuk, érdemes elvégeznünk.

Visszakeresés szempontjából két dolog kritikus: az egybe- és különírás, valamint a hosszú és rövid betűk. Az egybe- és különírás helyesírási szabályozása kissé laza, célszerű tehát, ha minden eshetőségre felkészülünk. (Annál is inkább, mert ezeket a hibákat a helyesírás-ellenőrző programok sem tudják kiszűrni.) A hosszú és rövid betűk megkülönböztetésének esélyét pedig rontja az a dogma, hogy a magyar helyesírás a beszélt nyelvet tükrözi. Ez ugyanis legfeljebb tendenciájában igaz. Kötve hiszem, hogy az „egyet" valaki is képes lenne egy gy-vel kiejteni. Ezenkívül a kiejtés vidékenként, szakmánként és sok más módon ingadozik. Különösen sok baj van az í-vel, például a típus vagy a színes leírása rövid i-vel szinte típushiba. Ugyanakkor a hosszú és a rövid magánhangzó a képzett vagy ragozott alakoknál is gyakran váltakozik (tízes — tizedes). Még a jó nyelvérzékűeknek is gyakran kell a helyesírási szótár után nyúlniuk, hogy elkerüljék a helytelen írásmódot.

Az automatikus szöveg-visszakeresés nincs olyan jól megoldva, ahogy azt a prospektusok nekünk tálalják, de talán nem annyira borzasztó, ahogy a fentiek alapján esetleg azok gondolhatnák, akik még nem használtak ilyen programokat. Én minden fogyatékosságuk ellenére tulajdonképpen elég jól elboldogulok velük.

Farkas Ernő

Aknamezőn lépkedve

Helyesírási gerillaháború

Ezt a cikket csak az **ne** olvassa el,
aki eddig még minden számítástechnikai kifejezéssel azonosan írt formában találkozott;
akinek soha nincsenek kétségei egy szakszó helyes írásmódjáról;
aki megoldottnak tekinti a számítástechnikai szaknyelv helyesírási kérdéseit;
és aki az egész kérdéskörnek semmi jelentőséget nem tulajdonít.

A számítástechnikai szaknyelv helyesírásával kapcsolatban bizonyos határig támpontot adnak a magyar nyelv helyesírására vonatkozó általános szabályok. A gond az, hogy ennek a határnak már az innenső oldalán is van néhány „akna", a túloldalon pedig elég ingoványos a terep, és nincs hozzá sem térképünk, sem kalauzunk. A hivatalos nyelvészeti dokumentumok szerint ugyanis azokban a kérdésekben, amelyekre vonatkozóan az általános szabályzat nem ad eligazítást, a szaknyelveknek módjukban áll saját normáikat kialakítani. Ez a szemlélet sajnos minden szakterületről feltételezi, hogy megvannak az egységes szaknyelvi platform megteremtéséhez szükséges körülmények. Hát a mi esetünkben ez elég nagy tévedés!

Az írásmódban kialakulhat az egység megegyezéses alapon vagy diktált módon. A számítástechnikára azonban sem az egyik, sem a másik út nem jellemző. Még egyazon intézményen belül sem nagyon lehet egységet teremteni, mert aki egyszer fegyvert (programnyelvet, szövegszerkesztőt, kiadványszerkesztőt) kapott a kezébe, az rögtön egyszemélyes gerillacsapatnak képzeli magát, és meg akarja vívni külön kis harcát. Ő maga sem tudja ugyan, hogy ki és mi ellen (hiszen többnyire még a magyar helyesírás általános szabályainak is mellélő, és elszigetelten, egyedül durrogtat az erdőben), de a keletkező irományokba pillantva elképesztő helyesírási kép tárul elénk.

Valamivel jobb a helyzet, amikor a számítástechnikai írások könyvben, folyóiratban jelennek meg, mert olyankor többnyire belép a folyamatba egy műhely (például szerkesztőség, nyomda), amely az általa kialakított normákat bizonyos fokig érvényesítni tudja. De a különböző műhelyek között már szintén nincs szervezett egyeztetés. Azok is csak egymás ellen elszigetelten harcoló partizánalakulatok — több emberrel és nagyobb kaliberű fegyverekkel.

Elvileg ugyan semmi akadálya annak, hogy a gerillaegységek képviselői egyszer mégis közös tárgyalóasztalhoz üljenek, de erre jelenleg hiányzik a belső késztetés, hiányzik hozzá a közvetítő és szervező munkát vállaló intézmény (ENSZ), és legfőképpen nincsenek meg tételes listába szedve

a megtárgyalandó alapdokumentumok.

Magunk részéről az egyeztetés belső igényének felkeltéséhez legfeljebb lapunk egészével tudunk hozzájárulni, remélhetőleg nem a rossz, hanem a jó példa hatásmechanizmusát működtetve. A szervezés gazdájának megtalálásához e havi kiemelt témánk cikkeivel próbáltunk közelebb kerülni, de az ügy érdekében a legtöbbet talán mégis azzal tehetjük, hogy teret adunk a megtárgyalandó kérdések felvetésének és azok előzetes megvitatásának. Ez utóbbit nyomban el is kezdjük, s igyekszünk majd folytatni is, örömmel fogadva, ha bárki hozzáteszi a maga problémafelvetéseit és javaslatait, vagy vitába száll mások álláspontjával.

Az állománynév és környéke

Kezdjük az egyik leggyakoribb dilemmával, amelyre vonatkozóan a Magyar helyesírás szabályai semmilyen eligazítást nem adnak. Hogyan írjuk le az állományok teljes nevét? Csak annyi bizonyos, hogy a névben lévő pontnak szerepelnie kell, mert az a jelrendszer része, megkülönböztető utalás a fájlnévjellegre és annak szerkezeti felépítésre. Írásmódja ezen túlmenően azonban már nem egységes. Vegyük sorra a lehetséges megoldásokat:

— autoexec.bat
— AUTOEXEC.BAT
— Autoexec.bat
— Autoexec.Bat
— autoexec.BAT
— AUTOEXEC.bat

Leggyakrabban az első hárommal találkozhatunk, legritkábban az utolsó kettővel. A csupa kisbetűs és a csupa nagybetűs változat azért terjedt el annyira, mert az operációs rendszerek és a fájlkezelő programok csak ezt a két formát alkalmazzák — az egyik így, a másik úgy.

Addig ezzel a kérdéssel nem is kell senkinek foglalkoznia, amíg a parancsok szintjén mozog, mert a programok automatikusan átírják a nevet a saját ízlésük szerint, akárhogyan gépeljük is be azt. Szövegszerkesztésben viszont már „mindent szabad", s válaszút elé kerülünk: köznévként vagy tulajdonnévként kezeljük-e? Hm... Ami az iménti konkrét példát illeti, az bizony közelebb van a köznévhez, mert minden számítógépben ott van, magunk alakítjuk ki, nem fűződnek hozzá szerzői jogok, s tulajdonképpen köznévi értelemben használjuk: „Beírtam az elérési útvonalat az autoexec.bat-ba." Mintha csak azt mondanánk, hogy „új hengerfejet tettem a motorba".

Ezzel szemben az állományok másik (jóval nagyobb) csoportjába tartozók egyediek, szerzői joggal védettek, s az írásmódtól függetlenül inkább tulajdonnévi értelemben használatosak: „A program a *kedit.exe* állománnyal indítható el." Ez a kis kezdőbetű ellenére inkább azzal az autós mondattal analóg, hogy: „A hegyi túrához érdemes *Pirelli* gumit felrakni." Annál is inkább, mert a *Kedit*, mint az egész programcsomag elnevezése szabályos tulajdonnév.

Az persze teljes káoszt okozna, ha egyenként kellene eldönteni a fájlokról azok nyelvi természetét. (Talán nem is mindig lehet!) Akkor pedig mi a megoldás? Egy kalap alá kell venni minden fájlnevet. Igen, de melyik változat szerint?

1. Az **autoexec.bat** változat azzal jár, hogy minden fájlnevet köznévnek tekintünk, ami tartalmilag azért nem szerencsés, mert az állományok többsége mégis inkább tulajdonnév. Felsorolásokban ez nem okoz zavart, de mondatban elhelyezve ellentmondásos, nehezítheti a megértést.

2. Az **AUTOEXEC.BAT** írásmód a tulajdonnévi bizonytalanság elfedése a kiemelés eszközével. Nem indokolt (és nem is esztétikus) a szavakat végig nagybetűsen írni, az csak a valódi betűrövidítéseknek jár, amikor nem „kimondjuk" a szót, hanem „lebetűzzük".

3. Az **Autoexec.bat** forma gyengéje az, hogy egyes fájlnevek inkább köznévi szóként használatosak, de talán ez az a pont, ahol a sokféle szempontnak mégis kompromisszumot lehet kötnie.

4. Az **Autoexec.Bat** írásmód feltehetően abból fakad, hogy a pont után nagybetű kívánkozik, annál is inkább, mert az csak a véletlen műve, hogy az elhatárolás jelölését a programozók annak idején a pontra definiálták, de az bármi más lehetne, általános nyelvi logika szerint például szóköz. A fájlnév két fele egyébként is úgy viszonyul egymáshoz, mint a keresztnév és a vezetéknév, ahol a kiterjesztés megfelel a családnévnek, az első (lehetséges) 8 karakter pedig a személynévnek.

5. Az **autoexec.BAT** írásmódot helyesírási normává tenni nem sok értelme lenne, mellette szóló érvként csak a kiterjesztés vizuális kiugratásával elérhető gyorsabb azonosítás hozható fel.

6. Az **AUTOEXEC.bat** írásmód szándéka talán azonos az előző pontban említett változatéval, de hatása sokkal gyengébb, mert az első tag rendszerint hosszabb, s nagybetűs írás esetén minél több betűből áll, annál inkább csökken az azonosítás sebessége.

Kiterjesztett szárnyak

A fájlnévnek a kiterjesztést tartalmazó fele önálló életre (szárnyra) kelt, jelzős szerkezetek gyakori szereplőjévé vált. A Vírushatározóban gyakran előfordul például az, hogy a vírus milyen állományokat fertőz meg. Azt az írásmódot választottuk, hogy „.COM fertőző vírus, .EXE fertőző vírus". De ezzel kapcsolatban bőven maradtak bennünk kétségek.

— Az első a kiterjesztés előtti *pontra* vonatkozik. Hogyan lenne helyesebb írni: az orientáló funkciót betöltő ponttal vagy annak elhagyásával?

— A másik, hogy végig nagybetűs legyen-e a kiterjesztés vagy végig kisbetűs? Esetleg nagy kezdőbetűs?

— A harmadik gond a kiterjesztésnek és a jelzett szónak az összekapcsolása: maradjon-e szóközzel elválasztva, vagy legyen helyette kötőjel. Ha kisbetűsen végződik a kiterjesztés, akkor pedig írhatjuk-e esetleg egy szóba is?

Itt most nem megyünk bele a részletes elemzésbe. Mi az említett könyvben letettük ugyan a voksot, de az Alaplapban sem mindig ugyanúgy „szavazunk", és nem is vagyunk meggyőződve arról, hogy miénk a jó megoldás. Várjuk ez ügyben is mindazok véleményét, akikben kialakult egy akár szakmai, akár nyelvészeti érvekkel alátámasztott állásfoglalás. Mankóként ehhez egy adott példával felsoroljuk, hogy mit is kell mérlegelni: .EXE fertőző, .EXE-fertőző, .exe fertőző, .exe-fertőző, .exefertőző, .Exe-fertőző, .Exe fertőző, .Exefertőző, EXE-fertőző, EXE fertőző, exe-fertőző, exefertőző, exe fertőző, Exe-fertőző, Exefertőző, Exe fertőző.

A bemutatott 16 változat közül kellene kiválasztani és használatra javasolni a legelfogadhatóbbat. A jelenlegi helyzetben ugyanis a szakmai anyagokban szinte mindegyik előfordul, ami bizonyára csak jelentéktelen megértési problémát okoz, de akkor sem válik a magyar számítástechnikai szakma nagy dicsőségére. Ráadásul éppen egy olyan szakmáról van szó, amely a programszövegek szintaxisában viszont teljes precizitást követel meg, s egyetlen karakter elgépelése is (lásd szovjet Marsszonda) hatalmas károkat tud okozni. A programozás által követelt pontosságból valami töredéknyi igazán kisugározhatna a szaknyelvre is!

Faklen Pál

Nyelvi morfondírozások

Multimédia vagy multimédium?

Az elmarasztaló glosszák időnkénti felbukkanása és a sajtó egy részének „médiumplántáló" igyekezete ellenére, a tömegtájékoztatási eszközökben, szakmai kiadványokban és a köznapi beszédben is tovább tartja hadállásait a média kifejezés — nem tisztelvén kellőképpen a latin többes számot. Miután a médium kontra média per a multimédián és több más szakkifejezésen keresztül közvetlenül érinti a számítástechnika nyelvét is, szándékaink szerint sorozatként indított nyelvi morfondírozásaink első darabjaként ezt a témát választottuk.

Aki netán még nem találkozott volna a *média* szót támadó vagy védő nyelvészkedő vitacikkekkel, annak is észre kellett vennie a sajtó körüli csatározások kétféle szóhasználatát: az egyik tábor médiumokat, a másik médiákat emleget, de nem politikai elkötelezettség alapján, hanem tisztán nyelvi megfontolásokból. Vajon melyik félnek van igaza? Az egyik szemléletmód latin nyelvi szabatosságot kér számon a magyar nyelvtől, a másik elfogadja a formális logika szabályait időnként nem tisztelő nyelvfejlődés tényeit.

„Szellemidézésre foglalt!"

Mindenekelőtt talán azon kellene elgondolkodni, hogy a spontán szóhasználat a kommunikációs eszközökre miért a *média* alakot részesítette előnyben, hiszen a fogalom megjelenésekor már ott állt készenlétben az egyes számú latin alakból meghonosodott *médium* is. Hát igen, éppen ez az első bökkenő. A *médium* szóra évtizedekkel előbb rátelepedett egymáshoz elég közel álló három másfajta jelentés:

— Hipnotizált és hipnotizálható személy.

— Valamely ügy eszközként felhasznált szereplője.

— Spiritiszták túlvilági közvetítő alanya. („A" médium.)

Emellett a médium szónak a latinban meglévő általános tárgyi, illetve fogalmi értelmű *közvetítő eszköz* jelentése nálunk nem akart gyökeret ereszteni. És akkor jött a média... Annak meg sikerült! Hogy is van ez?

A magyar nyelv új fogalmak nyelvi köntösét keresve nem nagyon szereti a már lefoglalt jelentésű szavakat. Lehet vitatkozni azon, hogy ez jobb vagy rosszabb annál az angol szokásnál, hogy ott a szó előszeretettel aggat magára megannyi jelentéstartalmat, de tény, hogy a mi nyelvünk kifejezetten szenved, ha időnként ilyesmivel traktáljuk. Talán a *médium* szó korábbi tartalmának mindig *személyre* utaló jellege is taszította az egészen más szférákban mozgó fogalomrendszert.

Annyi bizonyos, hogy bár a *média* szó sokáig csak szűkebb szakmai körben volt használatos, mégis általános jelentésbeli különbség alakult ki a kettő között. Egészen mást jelent a *médiumkutatás,* mint a *médiakutatás,* vagy a *kísérleti médium,* mint a *kísérleti média. Médiumtörvényt* is lehetne hozni (a spiritiszta médiumok működéséről), és az valószínűleg könnyebben is készülne el, mint a *médiatörvény.* Az okkultizmus felszínre törése köznapi beszédünkben is teret adott a *megfelelő médium kiválasztásának*, amit az érintettek (bármelyik oldalon) aligha szeretnének összekeverni a *megfelelő média kiválasztásának* műveletével. Szemantikai oldalról nézve tehát a *médium* és a *média* eltérő jelentések hordozójává vált, s ez önmagában is megerősítette a szóalakok elkülönülését.

A tájékoztatási eszközöknek *média*ként való megnevezését általában nem is jelentéstani, hanem (latin) nyelvtani megfontolásokból támadják, amiért a latinul már többes számú értelmet nyert *média* alakot helytelen módon egyes számban értelmezzük, és a *médiák* formával duplán tesszük többes számba. Csak hát a *média* már nem *media*! A más nyelvekből átvett szavak a befogadó nyelvben mindig önálló életre kelnek. Elveszítve eredeti formájukat, alkalmazkodnak a befogadó nyelv hangrendszeréhez (mert másként nem tehetnek), s elveszítik a ragozásukra, képzésükre vonatkozó korábbi nyelvtani szabályokat, hiszen minden nyelvnek önmagával konzisztens rendszert kell alkotnia, s a jövevényeknek nem a küldő nyelv szabályai szerint kell viselkedniük, hanem a befogadó nyelvé szerint.

Dupla vagy semmi

A *média* szó a magyar nyelvi környezetbe átlépve és ott meghonosodva már egyáltalán nem képes érzékeltetni eredetileg többes számú mivoltát, ezért sem okoz zavart, amikor *médiákat* emlegetünk. Ha magyar beszédünkből megpróbálnánk kiiktatni mindazokat a nyelvi formákat, amelyek más nyelvekből származnak és amelyeknek általunk történő használata nyelvtanilag nem követi az átadó nyelv szabályait, akkor valószínűleg lépten-nyomon megakadna a beszédünk. Jövevényszavaink túlnyomó többségére éppen az jellemző, hogy az eredeti szóalakhoz képest módosulva (sőt csonkán vagy megtoldva) honosodnak meg, mert csak ezen az áron tudnak beilleszkedni hangrendszerünkbe, nyelvtanunkba. Lehet, hogy ez illetlenség az átadóval szemben, de ettől még tény, sőt törvényszerű folyamat, amely a többi nyelvben is hasonlóképpen játszódik le.

A *média* szó „viselkedése" nem valami rendkívüli jelenség, de legközelebbi rokonai már annyira szervesen beilleszkedtek, hogy eszünkbe sem jutna származásukat — és többes számú alakjukat — firtatni. Íme néhány a változatlanul vagy magyaros írásmódban átvett szavak közül, amelyek eredeti latin többes számú alakját egyes számban használjuk, és gyanútlanul tehetjük saját nyelvi logikánk szerint

„még egyszer" többes számba: *nulla, opera, orgona, széria, váza, tufa, textília, fekália, parazita.* Ebből a felsorolásból nem lóg ki a média.

A média kifejezés egyes számú főnévként való elfogadását valószínűleg elősegítette egy formális analógia is (ami persze egészen más nyelvi és nyelvtani jelenség), az ugyancsak egyes számú alakban „-ia" végződésű, többnyire görög eredetű és elvont vagy gyűjtőfogalmat jelölő szavak nagy száma és gyakorisága. (Anomália, analógia, akadémia, biblia, frekvencia, fúria, mánia, memória, melódia, komédia, energia, kalória, fantázia, filozófia... és a sokféle -ológia.)

Rébuszokban beszélve

A vitákban a médium-pártiak időnként európaiságunkra szoktak apellálni. Pedig a *média* szó átvételében talán a nemzetközi gazdasági vérkeringésbe való bekapcsolódásunk is nagy szerepet játszott. A szakmai kifejezéseket leginkább orientáló angol nyelv ugyanis a saját testében is idegen *media* többes számú alakot széles körben terjeszti, az egyes számú *medium* alakot pedig csak speciális esetekben, egészen ritkán használja. Szóösszetételekben, jelzős szerkezetekben mindig a többes számú forma szerepel (*media research* = médiakutatás, *media planning* = médiatervezés, *media owner* = médiatulajdonos stb.), akkor is, ha annak magyar megfelelője egyes számú szóalakot követel meg. Mindez a *média* kifejezés egyes számú értelemben való meghonosodásának malmára hajtotta a vizet.

Egyre többször találkozhatunk azonban a *média* többes számú értelemben való használatával is. Pedig a magyar nyelv a média szót aligha tudná többes számúként kezelni, hiszen akkor olyan mondatokkal kellene megbirkóznunk, hogy például „A vidéki média *foglalkozzanak* inkább a helyi ügyekkel!". Az ige gyorsan leleplezi az össze nem illőséget, birtokos szerkezetet használva azonban áthatolhatatlan fogalmazási homály keletkezik. Ha azt mondjuk, hogy „a média irányításában személyi ellentétek vannak", az vonatkozhat a tájékoztatási eszközök összességére éppúgy, mint egyetlen újságra vagy a televízióra, s az sem biztos, hogy a szövegkörnyezet alapján egyértelműen el lehet jutni a helyes megfejtésig. Pedig, ugye, Kosztolányitól már megtanultuk, hogy az olvasót bizonytalanságban hagyni egy pillanatra sem szabad!

A címben szereplő *multimédia* is okozhat egyeseknek némi fejtörést. Egyrészt nagyon valószínű, hogy e szó nálunk ebben a formában terjed el, hiszen tömör, jól hangzik, beilleszthető a magyar nyelvbe, és az ékezetet leszámítva az egész világon így használják. Helyette mást javasolni (és elfogadtatni!) elég nehéz lenne. A *multimédium* bevezetése a médium-pártiakat saját okfejtésükkel állítaná szembe, mert a multi az összetétel másik felére vonatkozóan sokszorosat jelent, akkor pedig az hogy állhatna egyes számban? A *multimédiumok* sem megoldás, mert a fogalomhoz egyes számú főnevet is kellene alkotni, a csak többes számú (a magyarban egyébként is igen ritka) alak kezelhetetlen lenne. Ráadásul a multimédia olyan gyűjtőfogalom, amely csak egészen kivételesen kaphat többesjelet, tehát ritkán lenne alkalom a „többes-többes" miatti háborgásra.

Mit mond az írás?

Valószínűleg mindenkinek bele kell törődnie a *média* szó (vagy csak a multimédia?) meghonosodásába, annak összes nyelvhasználati következményével együtt. Erre a pontra a nyelvészek már korábban eljutottak, s érdemes az MTA Nyelvtudományi Intézete által 1985-ben megjelentetett kétkötetes Nyelvművelő kézikönyvben erről a témáról közölt szócikket teljes egészében idézni:

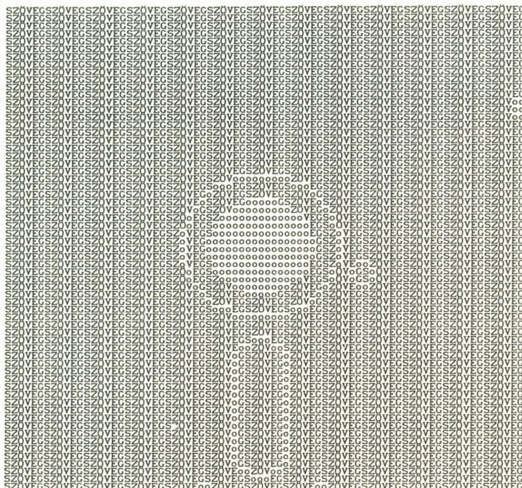
„Mivel a latin eredetű *médium* ma elsősorban szuggerálható, hipnotizálható, ill. könnyen befolyásolható személyt jelent, s csak másodsorban közvetítő eszközt, tényezőt, közeget, bizonyos társadalomtudományok, a híradástechnika és a reklám szaknyelvében *angol* mintára elterjedt a szó latin többes számú alakja, a *média*. Összefoglaló megnevezése a tömegkommunikációs, ill. reklámeszközöknek (sajtó, rádió, tv; plakát, hirdetés, reklám stb.).

Gyakran többes számban is használják: *médiák.* Ez voltaképpen szabálytalanság, mivel a már amúgy is többes számú szóalak kapott újabb többesjelet (mint pl. az *Alpesek, Beatlesek*): a legjelentősebb *médiák*; *reklámmédiák*; stb. A szó idegen volta és a *médium*-tól való jelentésbeli elkülönülése folytán azonban ma már általában nem ismeretes, hogy latin többes alakról van szó, s így ez a „szabálytalanság" aligha lehet zavaró." (II. kötet, 111. oldal.)

Tegyük hozzá kiegészítésként, hogy azóta a média szó jelentéstartalmát meghatározó sajtó, reklám és hírközlés mellé a számítástechnika is gyors ütemben felzárkózik, bár a köznyelvbe ez még nem nagyon szivárgott át. (Talán a multimédia lesz az első „betolakodó".)

A vita persze a nyelvészeti szakvéleményektől függetlenül a gyakorlati nyelvhasználat „népszavazásán" dől majd el, s ha a *média* jobban megfelel a magyar nyelv belső törvényszerűségeinek, akkor bármilyen nemes szándék is vezeti a másik felfogás képviselőit, hiába „kötik a médiumot a karóhoz".

Faklen Pál



SZÖVEGKÖRNYEZET

Akár életfogytiglanit is kaphat!

100 szónak mennyi (lesz) a vége?

Igaz ugyan, hogy nyelvi kérdésekben nem népszavazásokon szoktak dönteni, de tulajdonképpen mindig a nép dönt. Még csak nem is (választott) képviselőin és (önjelölt) szószólóin keresztül, hanem a lehető legközvetlenebb módon: ahogy anyanyelvét használja. De azért nem árt néha megkérdezni a népet, hogy miként vélekedik bizonyos dolgokról. Az Alaplap, mint ezen összeállításával a számítástechnikai szaknyelvben érdekelt nép (önjelölt) szószólója, most ilyesmit kezdeményez. Játéknak tűnhet, amire olvasóinkat kérjük, de csak azoknak ajánljuk, hogy részt vegyenek benne, akik minden játékot nagyon komolyan tudnak venni.

„Pofonegyszerű" feladatot találtunk ki. (Olyan értelemben, hogy egyszerű volt kitalálni, és az lenne a pofon, ha nem érkezne rá elég válasz.) A résztvevőktől „mindössze" azt kérjük, hogy küldjenek be 100 szót az Alaplap szerkesztőségébe (1441 Budapest VIII., Reguly Antal u. 8.), lehetőleg mágneslemezre tett egyszerű szövegállományként. (A lemezt visszaküldjük.) A 100 tulajdonképpen 5 x 20, mert minden felsorolt kategóriában legalább huszat kell összegyűjteni. A listában szerepelhetnek önálló szavak, eltérő szóalakok és összetett kifejezések is, de valamennyit a számítástechnikai szaknyelvből kell meríteni.

A csoportosítás ismérvei:

1. A legjobban sikerült magyarítások.

2. Magyar írásmódban megtartandó idegen szakkifejezések.

3. Eredeti írásmódban megtartandó idegen szakkifejezések.

4. Száműzendő és magyarral helyettesítendő idegen szakkifejezések.

5. A szaknyelvből kigyomlálandó „vadmagyarítások".

Természetesen a 20—20 tétel csak a megoldás elfogadásának minimális követelménye, amelytől felfelé bátran el lehet térni. Nem beküldési feltétel ugyan, de a 4. és 5. kategóriában kívánatos, hogy a beküldő írja le a helyettesítésre általa leginkább alkalmasnak tartott szavakat és kifejezéseket is.

A beérkező anyag feldolgozásából elsősorban azt szeretnénk megtudni, hogy olvasóink mennyire ítélik meg velünk azonosan vagy tőlünk eltérő módon a szaknyelvi, szóhasználati kérdéseket. Másik célunk, hogy friss ötleteket merítsünk a számítástechnikai szaknyelv csiszolásához és az új (vagy akár régi) szakmai fogalmakat legjobban kifejező nyelvi megoldásokhoz. Lapunkban természetesen közreadjuk az összesítés eredményét is, mágneslemez mellékletünkre pedig — tanulságként és ötletadónak — szeretnénk a javaslatok özönét rátenni, amihez csupán annyi kell, hogy bennünket is elárasszanak a megoldások.

Hogy vállalkozókedvüket valamivel serkentsük is, a 25 legjobb anyag beküldője közül háromnak kisorsolunk egy nem akármilyen jutalmat: Örökös Tiszteletbeli Olvasók lesznek, vagyis az Alaplapot életfogytig ingyen kapják. (A nyertesek élete, illetve az Alaplap élete végéig!)

A beküldési határidő:

1992. október 30.

Szó és szöveg

Angol nyelven

Style counsel. (Riva Grammatik IV, Editor Software StyleWriter és RightSoft RightWriter helyesírás-ellenőrző szoftvercsomag.) PC User, 1990/6.

Digital Dharma. (Az ősi tibeti irodalom számítógépes rögzítése.) Electric Word, 1990/6.

Could WordPerfect Rhymer be finer? (Diákok nyelvtanulását segítő, 93.000 szavas amerikai-angol rímszótár a WordPerfect szövegszerkesztőhöz.) Byte, 1990/11.

The multilingual edge. (A számítógépes fordítás távlatai.) Byte, 1991/3.

Evaluating Computer Generated, Domain-Oriented Vocabularies. (Számítógépes szakszótárak gépi generálásának fejlődése — természetes nyelvű információ visszakeresése.) Information Processing & Management, 1990/6.

Text checkers: mind your grammar. (Hat szövegelemző, helyesírás-ellenőrző program bemutatása és értékelése.) What Micro?, 1991/7.

Can a grammar and style checker improve your writing? (Kilenc — Macintosh és MS-DOS rendszerekre írott — helyesírás- és stílusellenőrző programtermék bemutatása.) Byte, 1991/8.

Write on target: 15 writer's tools. (15 különböző típusú szövegellenőrző programcsomag bemutatása és értékelése.) PC Magazine, 1991/9.

Software reviews: Language translation program. (Az angolról spanyol nyelvre történő fordítást segítő, IBM kompatibilis gépen futó, könyvtári alkalmazásra ajánlott Spanish Assistant szoftver.) Library Software Review, 1991/május-június.

DM-Dialog. (Kísérleti nyelvi fordító rendszer.) Computer — IEEE, 1991/6.

Parallel access to an English dictionary. (Öt transzputerre elosztott angol nyelvű számítógépes szótár párhuzamos elérése.) Microprocessors and Microsystems, 1991/július-augusztus.)

Német nyelven

Überstzungsprogramme: man spricht Deutsch. (Idegennyelvű fordítóprogramok összehasonlítása és értékelése.) Computer Persönlich, 1991/2.

Einfach und gut: PC-Wörterbuch von Langenscheidt. (Langenscheidt német-angol/angol-német PC-szótár.) BIT — Büro und Informationstechnik, 1991/4.

Fehlerlos schreiben? (Helyesírás-ellenőrző és -javító programok tesztelése.) PC Praxis, 1991/5.

Rechtschreibprogramme unter der Lupe. (Helyesírás-ellenőrzővel rendelkező szövegszerkesztő programok összehasonlítása.) Chip, 1991/8.

Ways — der neue Windows-Knüller? (Ways für Windows online fordító és helyesírás-ellenőrző program tesztje.) PC Praxis, 1991/8.

Vorstellung: Fünf Übersetzungsprogramme. (Természetes nyelvű fordítóprogramok bemutatása.) PC Welt, 1991/11.

Rovatunkban hónapról hónapra, egymásra épülő ismeretekkel „bombázva" az olvasókat, betekintést adunk a mesterséges intelligencia kutatásának, az erre alapuló fejlesztésnek, alkalmazásnak a fejezeteibe. Leginkább átfogóan kapcsolódik össze az elmélet és a gyakorlat a szakmai köztudatban szakértő rendszerekként „tisztelt" MI-fogalom — és természetesen a konkrét produktumok — terén. Innen ugyan ismét elágazik majd a diszciplínákat is számba venni hivatott sorozat tematikája, de az e havi téma segít közel hozni az eddig tárgyalt anyagot azokhoz a szakemberekhez, akik a dolgokat szeretik saját „hagyományaikhoz" is visszavezetni — vagy fogalmazhatjuk fordítva: szívesebben nyújtózkodnak az újabb irányok felé, ha lábukat a saját „házuk táján" vethetik meg... Most következő cikkünk tehát részint felépítménye az eddigi írásoknak, melyek egyeseknek talán túlságosan elvontnak tűntek, részint alapul szolgál a kapott ismeretek célirányos rendezéséhez, vagyis a továbbiak — kinek-kinek a maga szakterületén hasznosulható — befogadásához.

A tudás minden előtt!

A számítógépes programozás történetének egyes szakaszai az ott alkalmazott programstrukturálási stílusokkal, programozási paradigmákkal jellemezhetők. Az egyes szakaszok ugyanis rájuk valló módon válaszolnak azokra a szoftvertechnológiai kérdésekre, hogy például : milyen program „jól stukturált"; milyen adat- és algoritmusábrázolási módokat kövessünk; milyen technológia (elmélet, módszertan, eszközkészlet) áll az alkalmazásfejlesztők rendelkezésére, stb. Az alábbiakban ilyen megközelítéssel vázoljuk a szoftvertechnológia — számos egyéb szempontból tekintve is vizsgálható — fejlődésének három legfontosabb szakaszát, bővebben tárgyalva a tudásalapú technológia alapkérdéseit. Az ismert — jól ismert — fogalmak és összefüggések mondhatni „visszaköszönnek", de most a viszonylatok megváltozására kell igazán figyelni.

Algoritmikus programok

Milyen szerkezetű egy hagyományos, algoritmikus program? Mindenki számára világos az 1. ábráról leolvasható képlet, amely szerint egy ilyen program kódja a programvégrehajtáskor passzív adatrészből, és az ezen dolgozó — vagyis aktív — algoritmusrészből áll. Az adatok között kiemelt szerepük van az input és az output adatoknak.

Egy hagyományos program célja tulajdonképpen egy függvény megvalósítása: "leképezni" az input adatokat a nekik megfelelő output adatokra (mint egyetlen adekvát megoldásegyüttesre). A leképezés módját, HOGYAN-ját a program algoritmusrésze fogalmazza meg. A programhoz szervesül a program belépési pontja(i)t tartalmazó rész is, amely(ek) révén indíthatja el a felhasználó a programot.

Az ilyen programok írására egyre magasabb szintű (algoritmikus vagy procedurális) programozási nyelveket hoztak létre, különböző programfejlesztési módszereket, rendszerfejlesztő környezeteket dolgoztak ki. Idővel azonban egyre több olyan programot kellett írni, amelyek nagyon sok, a háttértárakra kihelyezett adattal dolgoztak. Az 50-es években megjelent a fájl (adatállomány) fogalma. Számos fájlkezelő rendszer született, biztosítva a fájlokban tárolt adatok hatékony kezelését (beírás, módosítás, törlés, rendezés stb.). A fájlok, illetőleg a fájlkezelő rendszerek lényegében a programok adat- és algoritmusrészének kiterjesztéseként tekinthetők.

Adatállomány, -bázis, adatbázisrendszer, -alkalmazás

Az elmúlt 30 év során az egyszerű fájlkezelő rendszerek általánosításaként jöttek létre az adatbázis-kezelő rendszerek. Ezek egyszerre több felhasználó és többféle alkalmazói program számára központi helyen és redundanciamentesen tárolva biztosítják az adatok kezelését. Az adatbázis-kezelő rendszerek a nagy, (el)osztott adatbázisok kezeléséhez hatékony mechanizmusokat, kényelmes szolgáltatásokat nyújtanak. Program helyett itt adatbázis-alkalmazás(ok)ról beszélünk (ad hoc igényeket a végfelhasználó közvetlenül is megfogalmazhat). Mint a 2. ábra is mutatja, az adatbázis(ok) a program adatrészének, míg az adatbázis-kezelő rendszer az algoritmusrészének kiterjesztéseként funkcionál(nak).

Az adatbázis-technológia alkalmazása „kivette" az általánosan használt adatokat és kezelésüket az alkalmazói programok írói és a végfelhasználó kezéből. Az 1. ábrán látható képlet a következőképpen módosul: az adatállománnyal kibővített adatrész továbbra is passzív. Az algoritmikus rész helyét az adatbázis-kezelő rendszer, valamint a megengedett adatbázis-kezelő nyelven megírt alkalmazói programok foglalják el, azonban nem csak ezek. Egy adatbázis-alkalmazás ugyanis „kinyílt" a végfelhasználó felé: olyan — egymással kombinálható — szolgáltatásokat kínál, amelyekkel az feladatokat fogalmazhat meg, és adhat oda a rendszernek megoldásra. Még egy markáns különb-

ség: e szolgáltatásokkal mind a fejlesztő, mind a végfelhasználó lényegében nem valamely algoritmus megadásával, hanem leíró stílusban deklarálhatja a rendszer számára azt, hogy MIT, azaz milyen feltételeket kielégítő adatokat kíván outputként megkapni. Az algoritmikus programozás mellett megjelent tehát a deklaratív feladatleírási stílus is, amely jó néhány programozási nyelvre (például Lisp, Prolog) is jellemző, és amely természetes módon valósul meg, az ún. tudásalapú rendszereknél.

Az adatbázis-alkalmazások célja jellegében tér el a hagyományos programétól: az adott adatbázis(ok) nagyon sok alkalmazói programot szolgálnak ki, továbbá a rendszer nyitottsága a végfelhasználó felé sokféle (ad hoc) funkciót biztosít. Indítása nem explicit módon kijelölt belépési pont(ok)tól kezdhető, hanem egy lekérdező nyelvet használva tudja a felhasználó előírni, hogy MIT akar. Ami még lényegesen módosult: (például relációs adatbázis-alkalmazásoknál) egyetlen lekérdezés során mind az input, mind az output adatok lehetnek halmazok is, sőt, a lekérdező nyelven „programozó" halmazműveletekkel is manipulálhat.

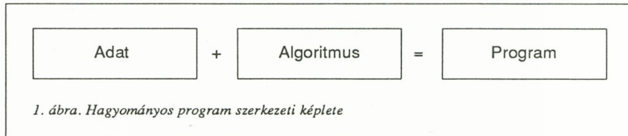
A programszervező, programtervező, programozó és végfelhasználó mellett kiemelkedik egy igen fontos új szereplő, az adatbázis-adminisztrátor. A működő rendszernek először lesz ilyen értelemben összetevője az ember:
adatbázisrendszer = adatbázis + -kezelő szoftver + -adminisztrátor

Időközben új szabványok és nyelvek jelentek meg: adatmodellező szabványok, adatbázisnyelvek/-szabványok (pl. Sequential Query Language, SQL), negyedik generációs nyelvek (4GL: pl. SQL*Forms), majd 1985 után a CASE (Computer Aided Software Engineering) technológia első változatai. Rohamléptekben készültek a különböző adatbázis-alkalmazások. 1988-ra már 400-nál több adatbázis-kezelő rendszer volt a piacon. A három klasszikus adatbázismodell (a hierarchikus, a hálózatos és a relációs) közül a 70-es évek közepétől a relációs modell terjedt el leginkább.

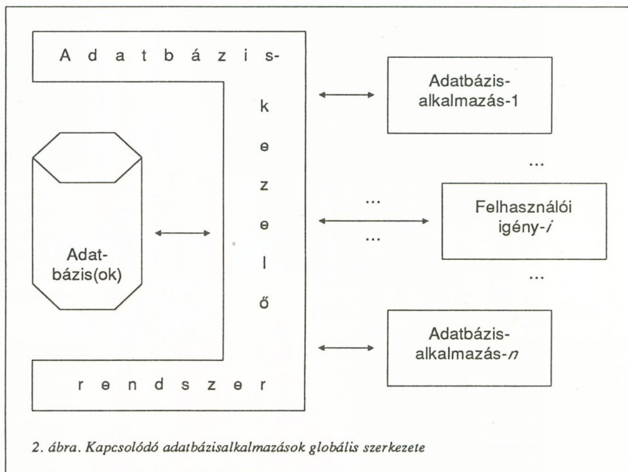
Az integráció hamarosan elindult: az adatbázis-kezelő rendszerek és programozási nyelvek egymásból hívhatósága ma már természetes igény, a piaci pozíciót erősen befolyásoló tényező.

Tudásalapú rendszerek

A MI-kutatások célja lényegében az intelligens emberi viselkedés megértése, s ennek alapján „intelligens" számítógépes programok létrehozása. Fel kellett ismerni: a problémamegoldás képessége lényegében attól függ, hogy mennyi, és milyen magasan kvalifikált minőségű specifikus ismeretanyagot képes mozgósítani a problémamegoldás során. A 80-as évek elején tehát különböző tudásábrázolási módszereket és (az így ábrázolt ismeretanyagot „működésbe hozó") problémamegoldó, következtető mechanizmusokat fejlesztettek ki — specializált tudásalapú rendszerek építése és működtetése céljából.

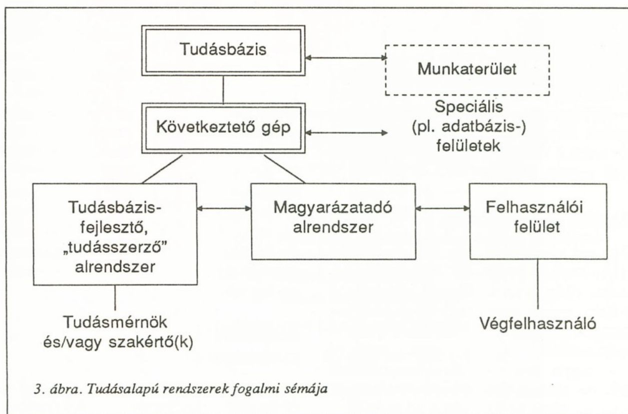


1. ábra. Hagyományos program szerkezeti képlete



2. ábra. Kapcsolódó adatbázisalkalmazások globális szerkezete

ismeretanyag — tudásbázis — áll rendelkezésre az adott tárgyterületről („a tudás elve"). Következésképpen egy program erőssége, intelligenciája is attól függ, hogy milyen sok, és milyen



3. ábra. Tudásalapú rendszerek fogalmi sémája

A tudásalapú rendszereknek a szerkezete sem a megszokott. Az alapelv itt: a tárgyterületi ismeretek különüljenek el a rendszer többi részétől, így a (hatékony megoldáskereső algoritmusokat tartalmazó) problémamegoldó ismeretektől is. Előbbit tudásbázisnak, utóbbit pedig következtető gépnek nevezzük (l. 3. ábra). Az ilyen szerkezetű rendszereket tudásalapú rendszereknek nevezzük. (Az első, bonyolult valós problémák megoldását célzó tudásalapú rendszereket szakértő rendszereknek hívták, azonban ez a megnevezés sok félreértés forrása volt, így ma egyre kevésbé „divatos".)

Az emberi problémamegoldási folyamatot elemezve mindenki megállapíthatja maga is, hogy bennünk jellemzően nem (előre beprogramozott) algoritmusok működnek, hanem egy probléma megoldásakor apróbb, önmagukban is értelmes tudáselemekkel, ismeretdarabkákkal manipulálunk: ökölszabályokkal (például: „Ha felkapcsoláskor nem ég a villany, akkor meg kell nézni, nem égett-e ki a villanykörte."), „tudáskeretekkel", frame-ekkel/sémákkal (gondoljunk például a magasabb rendű fajok tulajdonságait átörökítő állatrendszertanra, vagy egy számítógépes hálózat möködésének hierarchikus leírására). Ezt a működést nem tudjuk igazán jól közelíteni, ha megmaradunk a hagyományos algoritmikus programépítési szemlélet mellett.

Mint látni fogjuk, egy tudásalapú rendszer célja nem input és output adatok közti leképezések végrehajtása, nem is adott feltételeknek eleget tevő adatok kinyerése, hanem az, hogy adatokból (szak)ismeretekhez, szakértői szintű megállapításokhoz jusson el (például a beteg Morzsa kutya állapotát leíró adatokból megállapítsa, milyen kezelési módokat lenne célszerű alkalmazni a gyógyulás érdekében). Az output oldalon adat helyett információ jelenik meg: „Ezt és ezt tudtam kikövetkeztetni a (felhasználóval megbeszélt) aktuális szituációból kiindulva".

A tudásalapú technológia segítségével lehetővé válhat személyes vagy kollektív tapasztalatok, heurisztikus jellegű ismeretek számítógépes ábrázolása és működtetése. Ilyen programok tervezése ezért nyilvánvalóan el kell, hogy térjen a hagyományos programok (azaz egy adekvát megoldást kereső algoritmus programjának) tervezésétől. Kutató/felderítő jellegű (exploration-based) és modellközpontú tervezésre van itt szükség, amelynek jelenleg még mindig neuralgikus fázisa a szakspecifikus tudás megszerzése és rendszerezése (verbalizálás, elemzés, formalizálás stb.).

Képlet(es) játékok — nem képtelenség!

Az 1. ábrán látható képletre visszatérve, kizárólag a technológiai változások lényegének illusztrálása céljából, játszszunk el képzeletben a program szövetével. Az új igények figyelembevételével szervezzük át a programot a „strukturált programozás" korábban jól hasznosított alapelveinek megfelelően. A bal oldalon két programrész van; képzeletbeli átszerkesztésünk során induljunk ki először az algoritmusrészből, majd vegyük munkába az adatrészt:

1. játék

Daraboljuk fel értelmes, önálló részecskékre (például eljárásokra) az algoritmusrészt. Fogalmazzuk át e részeket deklaratív stílusban, HA AKKOR szabályokkal ábrázolt „ismeretdarabkák" formájában (például: „HA a primer hűtőkörből a szekunder körbe a hő átadásának mértéke elégtelen, ÉS a víz áramlási sebessége alacsony, AKKOR a rendszer állapota miatt baleset várható."). Megjegyzés: az adatok is ábrázolhatók így, például feltételrész vagy következményrész nélküli, elemi szabályokként.

Egészítsük ki ezt a készletet esetleg még további szabályokkal, majd helyezzük el ezeket egy tudásbázisnak nevezett rendszerkomponensbe, és működtetésüket bízzuk egy megfelelően kialakított végrehajtási mechanizmusra, egy ún. következtető gépre. Utóbbi egyrészt a következő feladatmegoldási lépésben végrehajtásra kerülő szabály kiválasztásával, másrészt a kiválasztott szabály végrehajtásával foglalkozik. Így egy ún. szabályalapú rendszerhez jutottunk.

2. játék

Az egymással kapcsolódó adatokat foglaljuk bele egy-egy adatstruktúrába: frame-be/sémába/tudáskeretbe. (Kapcsolt adatokként, pontosabban objektum-tulajdonság-érték hármasként ábrázolható például a „Morzsa szőre fehér" állítás.) A tulajdonságok öröklődése szerint jelöljük be e struktúrák hierarchikus kapcsolatait. Egészítsük ki az így leírt világot úgy, hogy az minél teljesebb legyen, és a kapott struktúrákat helyezzük el a tudásbázisban.

Ezután tördeljük fel és fogalmazzuk át az algoritmusrészt úgy, hogy az így kapott darabkák (például eljárások — vagy akár az előbb látott szabályok) értelmesen hozzákapcsolhatók legyenek frame-eink bizonyos elemeihez (például a hálózat egy pontján észlelt hiba azonnal hozzon működésbe egy, a hiba hatását továbbgyűrűztető eljárást, ún. démont).

Az így előállított „világleírás" működésbe hozatalához kapcsoljunk be egy megfelelően kialakított öröklődési, esetleg egy démon mechanizmust is. Ha pedig úgy döntünk, hogy a frame-ek közötti információforgalmat magukra a frame-ekre bízzuk (ilyen értelemben a frame-eket önálló objektumoknak tekintjük), az egymás közti üzenetváltások elősegítése céljából kapcsoljunk be egy megfelelő üzenetváltó mechanizmust is. És így tovább. Mint látható, a következtető gép itt nem egy helyen van koncentrálva, hanem „szét van osztva" (a 3. ábra csak elvi séma!). Az aktuális tudásbáziselem és a kívánt tevékenység kiválasztását a frame-ek adott hierarchikus kapcsolatai, valamint a frame-ek meghatározott pontjain elhelyezett információk (például az ezek hatására aktivizálódó démonok) vezérlik. A fenti mechanizmusok pedig végrehajtják a kiválasztott tudásbáziselemen a szituációnak megfelelő tevékenységet. Ilyképpen frame-alapú, illetve objektumalapú rendszerhez jutottunk.

Mindkét képzeletbeli játék lényege az volt, hogy kiemeltük a programból (annak procedurális kódjából) a tárgyterületi ismeretanyagot, és azt más szerkezetben ábrázolva belehelyeztük a tudásbázis nevű, új rendszerkomponensbe. Továbbá, az 1. esetben az algoritmusrészből kiemeltünk egy végrehajtó-következtető gépet, míg a 2. esetben azt mintegy szétterítettük átstrukturált, hierarchikus elrendezésű világunkba, segítségül adva egy öröklődési, egy démon, egy üzenetforgalmat biztosító stb. mechanizmust. Átstrukturált rendszereink úgy működnek, hogy a pillanatnyi szituáció szerint tesznek aktívvá egy szabályt, érvényesítenek valamely öröklődést, hajtanak végre egy démont, küldenek el vagy fogadnak egy üzenetet.

Maradjunk még továbbra is gondolati kísérletünknél. Amennyiben a fenti két átalakítást akképp végeztük, hogy a tudásbázisban, ahol csak lehetett, kódjuk helyett „nevén neveztük" a dolgokat, megteremtettük a lehetőséget arra, hogy a tudásbázis elemeivel dolgozó mechanizmusok szimbólummanipulációt végezhessenek. Ez „olvashatóvá" teszi a tudásbázist, valamint segít a

felhasználói párbeszéd, valamint a magyarázatadás során a képernyőn megjelenő (természetes nyelvű) közlések futás közbeni generálásában is. Ezért van az, hogy az ilyen rendszereket szimbolikus programokként is emlegetjük.

A hagyományos program gondolati átszerkesztése egyben választ ad arra is, hogy hol keressük az adat, és hol az algoritmusrészt. Itt már nincs szétválasztva az adatelérés és adatkezelés az eljáráshívástól. Adat csak az input oldalon van; a rendszer output oldalán megjelenő üzenet már információ (példánkban: kezelési tanács a kutya tulajdonosának vagy az állatorvosának), amihez jelentést természetesen a végfelhasználó, az ember ad. Összegezve megállapíthatjuk, hogy:

— a „HOGYAN adjuk meg a függvényleképezést realizáló algoritmust?" kérdésre választ adó algoritmikus programból egy,

— a „mely ismeretek megadásával tudjuk leírni a megoldandó feladatot" hozzáállással készített, leíró, deklaratív szemléletű, szimbolikus programot kaptunk, amely a „MIT kell tudni?" szemléletet fejezi ki.

Mint már említettük, így olyan szerkezetű programhoz jutottunk, amelynél a tárolt ismeretanyagot már nemcsak egy feladat, hanem egy egész feladatosztály esetén tudjuk mozgósítani (különösen, ha figyelembe vesszük azt, hogy átalakítás közben — a teljességre törekedve — ki is bővítettük a rendszerbe beépített ismeretanyagot).

Kreativitás és engineering

Az adatbázis-kezelő rendszerek megvalósításánál három fő elv érvényesült: nagyméretű adatbázisok nagyszámú felhasználó által egy időben történő használatának elve; adatfüggetlenség elve; a fizikailag különböző helyeken tárolt (ún. osztott, elosztott, megosztott) adatokhoz való hozzáférés elve.

Közös az adatbázisokban és a tudásbázisokban, hogy — az emberek problémamegoldását segítendő — számítógépben információt tárolnak. A jellegzetes különbség a tárolt információk típusában (adat — tudás/ismeret), valamint az általuk támogatott problémamegoldás eltérő jellegében van. Mély igazság van a következő, gyakran idézett képletben:

tudásbázis = kreatív adatbázis,

amelyet ma sok kutató/fejlesztő „szó szerint" értelmez. Nevezetesen arról van szó, hogy a (hatékonyan működő) adatbázis-kezelő rendszereket egy meghatározott irányban fejlesztik tovább: adatok mellett kifejezéseket, „ismeretforgácsokat" lehessen bennük tárolni, és azokat feladatmegoldásra mozgósítani (intelligens adatbázisok, objektumorientált adatbázisok — kibővített tudásalapú rendszerek).

Mint láttuk, a tudásalapú rendszerek a tárgyköri ismeretanyagot („tudást") explicit módon ábrázolják; ennek főbb előnyei:

— természetes módon lehet új ismeretdarabkákat beépíteni, a régieket módosítani — lehetővé téve így az ún. inkrementális rendszerépítést (annak összes előnyével és veszélyével együtt);

— mind a rendszer pillanatnyi állapotáról, mind a problémamegoldás lépéseiről kézenfekvő a magyarázatadás, az indoklás: „WHAT (MIT) tartalmaz a tudásbázis (például: mely szabályok foglalkoznak a hőátadással)?", „WHY (MIÉRT) tett fel a rendszer egy kérdést?", „WHAT IF (MI LENNE, HA) ezt és ezt válaszolnám?", „HOW (HOGYAN) következtette ki a rendszer az eredményt?", „WHY NOT (MIÉRT NEM) ez és ez lett az eredmény?" stb.;

— a számítógépben ábrázolt tárgyterületi ismeretanyag és (az erről való tudás, azaz) a metatudás explicit ábrázolása előfeltétele az ismeretek automatikus „megszerzésének", megfelelő tanulómechanizmusok kidolgozásának és beépítésének (amely alapkövetelmény a tudásalapú rendszerek következő generációjánál).

Ha tehát rendszereinket ellátjuk egy (természetes nyelvű, grafikus lehetőségekkel rendelkező stb.) felhasználóbarát felülettel, valamint (az embert utánozva) megfelelő magyarázatadási képességgel (legalább a WHAT, WHY és HOW lehetőségekkel), akkor egy újszerű programszervezési elvek szerint felépített, ún. tudásalapú rendszerhez jutottunk. Egy ilyen tudásalapú rendszer működését a felhasználó jellemzően nem belépési pont (vagy valamely lekérdező nyelven megfogalmazott keresési cél) megadásával indítja, hanem — mint intelligens beszélgető partnerrel — „konzultál" vele az aktuálisan megoldandó problémáról (adatot közöl vele, a rendszer felkérésére dönt egy adott kérdésben, magyarázatokat, indoklásokat kér a rendszertől stb.).

Specialisták és „konzíliumok"

Két új szereplő lép előtérbe: a tudásalapú rendszerépítésben jártas tudásmérnök (ismerettechnológus) és a szakismereteit átadni képes tárgyköri szakértő. A végfelhasználó is más, mint eddig. Nem „kezeli" a programot, hanem tanácsadó partnerként „megbeszéli vele" a problémát, a rendszer által adott indoklások figyelembevételével átgondolja a rendszer javaslatait, majd dönt (a döntés felelőssége az emberé!). Egy tudásalapú rendszer tehát intelligens információfeldolgozó rendszernek tekinthető. Bár a valóság ennél bonyolultabb, már az eddigiekből is látszik, hogy a tudásalapú technológia forradalmasította nemcsak a rendszerépítést, hanem az ember-gép kapcsolatot is.

Tudásalapú rendszerek kifejlesztésére és futtatására ma már sok keretrendszer, shell található a piacon; ezek segítenek az új technológia elterjesztésében, valamint egyszerű rendszerváltozatok, prototípusok elkészítésében. Az első shellek megjelenése után egy ún. technológiavezérelt, prototípus-centrikus rendszerépítési hozzáállás volt jellemző: adott shellhez, eszközkészlethez keresni megfelelő alkalmazásokat (rossz esetben: minden alkalmazást egy adott shell felhasználásával megvalósítani). Ebben a szakaszban sok prototípus készült világszerte, de igazán csak 20 körüli volt használatban (1988 végén a piac befagyni látszott).

Jelenleg egy egészségesebb, ún. „feladatvezérelt", modellvezérelt rendszerépítési szemléletmód kezd egyre jobban terjedni. Ennek lényege a következő: ismerjük meg előbb az adott feladat természetét, jellegét, majd ezután keressük csak meg a („hozzá legjobban illő") megvalósítási eszközöket.

Elemi igény itt is a korábbi technológiákkal való „együttélés": szükség van például műszaki számítások hagyományos nyelvű eljáráskönyvtárainak, grafikus eljáráskészleteinek stb. futás közbeni felkeresésére, nagy adatbázisokban tárolt adatokhoz való hatékony hozzáférésre, egymással együttműködő tudásalapú rendszerek egyazon számítógépes hálózatba kapcsolására stb. Mindezek megvalósításához intelligens komplex rendszerek szükségesek; ez pedig elemi igényként veti fel a hagyományos programozási iskolák, valamint az adatbázis-technológiák és a (még „éretlen") tudásalapú technológia integrálásának szükségességét.

Ebbe a vonulatba a hazai szoftverfejlesztők és szakértők elég hamar bekapcsolódtak. Az Alaplap következő számában néhány ígéretes munkáról olvashatnak a már megvalósult, működő szakértő rendszerek iránt érdeklődők.

Sántáné Tóth Edit

PADSor I.

Vége a mellőzöttségnek

„Ingyen PADS" akció keretében a komplett PADS (Personal Automatic Design System) alaprendszer limitált változatához juthattak hozzá az érdeklődők. A 6000 elemes alkatrészkönyvtárt is tartalmazó szoftver vizsgálata során — amelyről két részben számolunk be — kiderült, hogy a „kis rendszer" tényleg tudja mindazt, amit beharangoztak róla: igazán felhasználóbarát környezetben 60-70 alkatrészből felépített áramkörök gyártásra készre(!) tervezhetők vele.

A könnyen kezelhető, moduláris, PC bázisú professzionális PADS rendszernek először a layout-tervező (PADS-PCB), majd a sémaeditor (PADS-Logic) változata készült el. A fejlesztés újabb lépcsőfokát a PADS-2000 program jelentette, amely — a konkurens programcsomagokat megelőzve — PC-s környezetben nyújtja a munkaállomás alapú tervezőrendszerek legfontosabb szolgáltatásait. Ugyanakkor napjaink igényeinek megfelelően megszületett a rendszer Unix alapú, Sun Sparcstation munkaállomásokon futó változata is, ez a PADS-2000/UX.

A PADS megjelenése előtt az analóg áramkörök konstruktőrei úgy érezték, hogy „mellőzöttek", hiszen a tervezőprogramok inkább a digitális áramköri technológiának kedveztek. A PADS fejlesztői szakítottak ezzel a filozófiával. Így a szoftver számos olyan szolgáltatást tartalmaz, amely analóg és nagyfrekvenciás áramkörök tervezésére is alkalmas. Nem véletlen tehát, hogy a PADS rövid idő alatt közismert és közkedvelt lett a nyáktervezők körében.

Redundanciamentes

A PADS rendszer része az alkatrészkönyvtár, amely jól felépített szerkezetével szükségtelenné teszi a nagy mennyiségű adat és grafikus elem redundáns tárolását. Az alkatrészkönyvtár négy része az adatlap-, a szimbólum-, a PCB formakönyvtár, valamint a grafikai elemek könyvtára.

Az adatlapkönyvtár tartalmazza az elektromos elemek részletes paramétereit: az elem nevét, logikai családját, a variánsok (az azonos funkciójú, kivezetésű, elrendezésű elemek) listáját, valamint az elemen belüli azonos funkciók (tipikusan kapuk) számát, felcserélhetőségüket. Itt utalunk az elemhez tartozó rajzszimbólumra, a tokozásra, valamint azok alternatív megfelelőire. Rajzszimbólum esetén ez lehet a logikai kapu De Morgan-megfelelője, míg a tokozásnál a ténylegesen létező összes típus szerepel. Konkrét grafikus definíciókat azonban az adatlapkönyvtár nem tartalmaz.

A szimbólumkönyvtárban azok a grafikai jelölések szerepelnek, amelyek a kapcsolási rajz lapjain találhatók meg. Ennek elemeit grafikus támogatással a felhasználó is bővítheti, amelyhez egy általános célú grafikai könyvtár elemei közül válogathat.

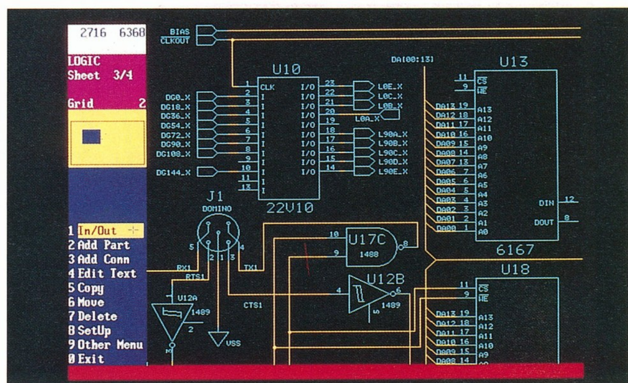
A PCB-formakönyvtárban találhatók meg az egyes tokozási típusok definíciói, a tok körvonala, a kivezetések helyzete, mérete, továbbá a forrszemek alakja. A formakönyvtár természetesen tovább bővíthető.

A grafikai elemek könyvtára olyan szimbólumokból áll, amelyeknek nincs elektromos tartalmuk. A szimbólumok a kapcsolási rajzon és a nyomtatott áramköri panelen egyaránt szerepelhetnek. A felhasználó kialakíthatja például cége emblémáját, amelyet a továbbiakban minden rajzon és terméken elhelyezhet.

A nagy helyigényű grafikus szimbólumok csak egy példányban szerepelnek az alkatrészkönyvtárban, így a tárolás „takarékos". A redundanciamentes tárolási eljárásnak köszönhetően a jelenleg 7400 különböző elemet tartalmazó alkatrészkönyvtár valójában 15-16 ezer alkatrészt jelent — mindössze 3 megabájtnyi lemezterületen.

Sémákban gondolkodik

A PADS-Logic modul a rendszer sémaeditora. Segítségével rendkívül egyszerűen szerkeszthetők meg a kapcsolási rajzok. A program 16 és 32 bites változatából 286-os gépen a 16 bites, 386/486-os gépen pedig a 32 bites



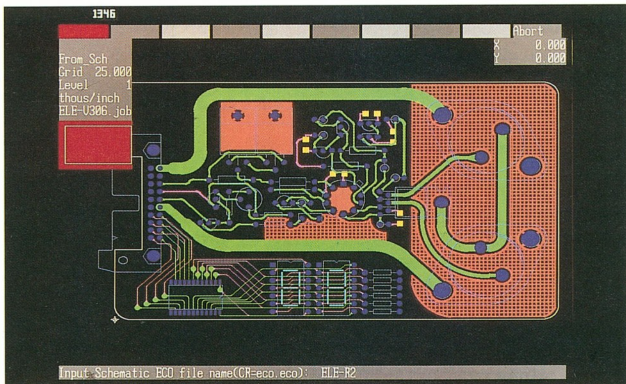
változatot installálják. Használata EGA vagy VGA videoadaptert, egeret, legalább 2-4 Mbájt memóriát igényel. Általában igaz, hogy a 32 bites verzió 4-5-ször több alkatrészt, összeköttetést kezel, mint a 16 bites változat.

A PADS-Logic program kezelői felülete azonos a rendszer többi moduljával. Kidolgozásakor alapelv volt az egér és a billentyűzet együttes használata, mivel így lényegesen gyorsabban érhetők el az egyes funkciók. A 16/32 bites változatban a képernyő bal oldalán/tetején látható a 10 elemből álló menürendszer, amely a billentyűzet 10 funkcióbillentyűjének felel meg. A menüt úgy alakították ki, hogy egy funkciót több helyről is elérhetünk ugyanazzal a billentyűvel. Például az F10 mindig kilépést jelent, a képernyő alsó sorában pedig szöveges információk vihetők be. Ahol alkatrész- vagy fájlnevet kell megadni, ott a * jel hatására egy listát kapunk a választható elemekről.

Nagyban megkönnyíti a program használatát, hogy a grafikus szimbólumok szerkeszthetők. Például ha meg akarjuk változtatni valamelyik szimbólumot, vagy egy újat akarunk létrehozni, ezt a Logic programból való kilépés nélkül tehetjük meg.

Figyelemre méltó a képernyőn látható szálkereszt helyzetérzékeny kiválasztása. Így a mozgatás vagy törlés parancs használatakor nem kell külön meghatározni, hogy milyen típusú objektumon akarjuk a műveletet elvégezni, elegendő ezt a szálkereszttel kiválasztani.

A felhasználó által definiálható makrók a többször ismétlődő műveletek egyszerű és gyors végrehajtását teszik lehetővé. Az ún. ideiglenes makró csak a program egyszeri futása alatt használható, míg a különböző billentyűkhöz rendelhető állandó makrók elmenthetők, bármikor felhasználhatók, utólag megváltoztathatók.



Mivel a korszerű, nagy integráltságú elektronikus kártyák kapcsolási rajzai rendszerint olyan mennyiségű alkatrészt tartalmaznak, hogy azok egyetlen elfogadható méretű rajzlapon sem jeleníthetők meg, a tervet rajzlapokra kell bontanunk. Ez a felbontás lehet hierarchikus vagy mellérendelt. Az előbbi esetben a fő (a hierarchia csúcsán álló) rajzlapon elhelyezett modulok egy-egy önálló lapot jelentenek, míg az utóbbinál az egyes lapokat egymás mellett kell elképzelnünk. A program nem korlátozza a hierarchikus felépítés szintjeinek számát.

A PADS-Logic egyik nagyon fontos jellemzője — ami korábban csak a munkaállomásokon működő rendszerek sajátja volt — a kártyaorientáltság. A hasonló sémaeditorok rendszerint laporientált működésűek. A felhasználás szempontjából fontos megkülönböztetni a két elvet. A laporientált sémaeditorok logikai egysége a rajzlap, így egyidejűleg csak az aktuális lapon levő alkatrészek és kapcsolatok töltődnek a számítógép memóriájába. A kártyaorientált PADS-Logic viszont együtt kezeli a teljes terv adatbázisát. Így lehet például egy jelet vagy egy áramköri elemet — bármelyik lapon is legyen — könnyedén megtalálni.

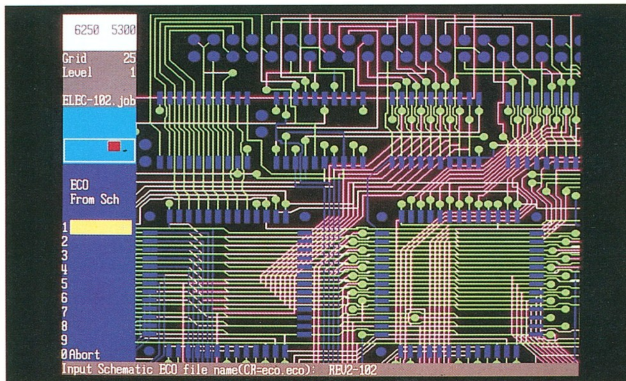
Ugyancsak nem kell kilépni a szerkesztőprogramból ahhoz, hogy elvégezzük a terv logikai helyességének ellenőrzését és az automatikus alkatrész-sorszámozást.

A PADS-Logic menüjébe beillesztették azokat a programmodulokat is, amelyek a kapcsolási rajzból alkatrész- és összeköttetési listát állítanak elő. Az alkatrészlista tetszőlegesen formátumozható, ízlésünknek megfelelő címeket, tételeket definiálhatunk — mindezt magyarul is.

A teljes tervadatbázis (vagy annak bizonyos részletei) kimenthetők vagy betölthetők ASCII-formátumban is. Ez megkönnyíti az illeszkedést olyan rendszerekhez, amelyekhez ezt a PADS nem biztosítja. Ugyanakkor az ismertebb tervezőrendszerekkel „közvetlen a kapcsolat".

Manuális és automatikus

A kapcsolási rajz hibátlan elkészítése és ellenőrzése után hozzáláthatunk a tényleges nyomtatott áramköri lap tervezéséhez. Ebben segít a PADS-PCB modul, amely olyan interaktív szerkesztőprogram, amelyet felkészítettek a kü-



lön megvásárolható automatikus alkatrész-elhelyező, huzalozó- és a gyártási kimenetet előállító modulok integrálására. A PADS-PCB — ezen modulok nélkül — az alkatrészek manuális elhelyezését és huzalozását végzi el. Kezelői felülete megegyezik a Logic modullal.

A PADS-PCB kapacitása legfeljebb 250 ekvivalens IC méretű kártyák kezelését teszi lehetővé. A PCB modul fontos feladata a kártya fizikai méreteinek definiálása. Segítségével a pontos kártyakontúr inchben vagy metrikus egységben is létrehozható.

Az alkatrészek elhelyezéséhez először az ún. pozicionálási mátrixot kell létrehozni. A mátrix pontjai szolgáltatják az alapot az alkatrészek automatikus elhelyezéséhez. Az alkatrészek egy csoportja azonban igényli a kézi elhelyezést. Ilyenek a kártyán a rögzített helyű csatlakozók, élcsatlakozók, kapcsolók, potenciométerek, LED-ek, és minden más olyan elem, amelynek igazodnia kell valamilyen külső geometriához. Ezek az alkatrészek a PCB modul segítségével pozicionálhatók végleges helyükre. Bizonyos analóg vagy nagyfrekvenciás alkalmazások is szükségessé tehetnek olyan különleges alkatrészelrendezést, amelyet a tervező csakis manuálisan valósíthat meg.

A PCB modul hatékony segédeszközökkel támogatja az alkatrészek kézi elhelyezését. Különféle csoportosításban automatikusan adagolja az elemeket, mozgatás közben folyamatosan mutatja az új elem kivezetéseinek legrövidebb bekötését (rubber banding). Az alkatrészek forgathatók, tükrözhetők, kicserélhetők más tokozású változatra. A kötött helyű alkatrészek kézi elhelyezése után elindíthatjuk a PADS-Place programot, amely a hátralevő alkatrészek automatikus elhelyezését végzi el. Amennyiben valamilyen ok miatt elégedetlenek vagyunk az eredménnyel, utólag természetesen kívánságunknak megfelelően módosíthatjuk is az elrendezést.

Az alkatrészek elhelyezése után a PCB modul kézi huzalozásra ad lehetőséget, ami jól jön a kritikus vonalvezetésű jelek útvonalának kialakítása során. A jelek eközben tetszőleges szögben vezethetők, de csak merőleges vagy 45 fokos vonalvezetés is előírható.

A kézi huzalozás közben folyamatos információt kapunk az aktuális jelről. Meglévő jelvezetéket vagy annak egy részletét is szerkeszthetjük, mialatt hozzáadhatunk vagy törölhetünk töréspontot a jel útvonalából, áttérhetünk más rétegre, illetve vonalvastagságot válthatunk.

Átjárható

A Logic modullal kialakított kétirányú kapcsolat révén a kapcsolási rajzon végzett utólagos módosítások a kártyatervre átvihetők, s a kártyatervezés közbeni változtatások (alkatrész-átnevezés, kapu- és kivezetéscserék) is visszajelölhetők a kapcsolási rajzra. A PADS rendszerben ECO (Engineering Change Order) kapcsolatnak nevezik ezt a lehetőséget, amely minden esetben biztosítja a kapcsolási rajz és a kártyaterv logikai összhangját.

Az igazán nagy méretű kártyák tervezői a PADS-Large modullal egyfajta teljesítményfokozást érhetnek el, segítségével a PADS-PCB határadatai mintegy másfélszeresre növelhetők. Működtetéséhez a PCB modul memóriaigényén túl további 1 Mbájt EMS memóriára van szükség.

Lóth Tamás

INFORMÁCIÓKÉRÉS: 20 ▼

Akik szeretik az egyszerűt, figyeljenek!

Csaknem Basic

Az #539 számú SolarSoft lemezen található Asic egy Basic programozási nyelv IBM PC-re. (Az Asic elnevezés eredete: „Its Almost Basic", ami magyarul csaknem Basic-nek fordítható.) Integrált editort tartalmaz, ebből szerkeszthetjük, fordíthatjuk, tesztelhetjük programjainkat, de természetesen saját kedvenc editorunkkal is dolgozhatunk, DOS-ból is fordíthatunk.

Az Asic a Basica és a GWBasic nyelvek alapján készült, 80-nál több Basic utasítást támogat, integer és string változókat, stringsorozatokat kezel. Fordítása gyors, az általa generált kód tömör. Az Asic utasítások csaknem mindegyike megtalálható a Basica és a GWBasic nyelvekben is.

Használatához 305-405 kbájt szabad memória szükséges. Maga az Asic fordító 305 kbájt memóriát igényel, az editornak elég 200 kbájtnál valamivel kevesebb. 405 kbájt akkor szükséges, ha az integrált editorból akarunk fordítani és hibát keresni.

Az Asic nyelv szabályainak megfelelő programok mindenki számára ismerősek lesznek, aki látott már Basic programot. Az első szembetűnő különbség az, hogy itt nem sorszámozunk. Nézzünk egy rövid programrészletet:

GWBasic/Basica

```
10 I=I+1
20 PRINT I
30 IF I<100 THEN 10
```

Asic

```
ADDONE: I=I+1
PRINT I
IF I<100 THEN ADDONE:
```

Mindkét program az 1 és 100 közötti számokat jeleníti meg a képernyőn. Az Asic azonban beszélő címkenevet használ a Basic sorszámok helyett. Ezek a címkék akár 80 karakter hosszúak is lehetnek. Vegyük észre azt is, hogy az Asic PRINT és IF utasításait nem kötelező beljebb kezdeni, ez csak a program olvashatóságát könnyíti meg! Írhatnánk összesen ennyit:

Asic

```
ADDONE:I=I+1
PRINT I IF I<100 THEN ADDONE:
```

A hatás ugyanaz lenne. A változónevek is lehetnének hosszabbak, egészen 80 karakteresig. Ugyanez vonatkozik a stringkonstansokra is.

Az Asic programok bevitele a lehető legegyszerűbb dolog. Soronként begépeljük, az egyes sorokat ENTER-rel zárjuk. Integrált környezetben a képernyő 2-24 soraiba írhatunk. A felső sor mindig a menü, a legalsó pedig a státusz-sor.

Induláskor négy menüből választhatunk: File, Edit, Compile és Run. Nevük önmagáért beszél. Run választás esetén nemcsak futtathatjuk, hanem tesztelhetjük is (az Asic debugerrel) programjainkat. Az alsó menüszinteken egyes műveleteket kivettek a menüszerkezetből, ezeket funkciós billentyűkhöz rendelték.

Online help segíti a felhasználókat. Asic-kulcsszavakat, ezek szintaxisát, hatásukat; az egyes menüpontokkal kiválasztható tevékenységeket jelenít meg a képernyőn.

A File menü állományok megnyitására, módosított tartalom mentésére (esetleg más néven) szolgál. Ebből a menüpontból tudunk visszalépni DOS szintre.

Edit menüben a szövegszerkesztésnél megszokott műveleteket végezhetjük: blokk-kijelölés, törlés, áthelyezés, másolás, karaktersorozat keresése, helyettesítés, ismételt keresés. Ami már több ennél: itt indíthatjuk a fordítást, itt állíthatjuk be a fordító kapcsolóit, köztük a nyomkövetést is. A fordítás megkezdése előtt a rendszer természetesen elmenti az aktuális forráskódot.

A státusz-sor editor módban a megnyitott állomány neve mellett '*'-gal jelzi azt, hogy már módosítottunk valamit, de nem mentettük el az új tartalmat. Megjeleníti az aktuális sor állományon belüli sorszámát és az ablakon belüli sor-oszlop kurzorpozíciót is. Editor módban a billentyűzet a megszokott módon viselkedik, az esetleges eltéréseket a kézikönyv tartalmazza.

Run menüből indíthatjuk a sikeresen lefordított, már hibátlannak tűnő programokat, ki/be kapcsolhatjuk a nyomkövetést, az utasításonkénti végrehajtás-felfüggesztést, módosíthatjuk az egyes változók értékeit, megjeleníthetjük a kijelölt változók aktuális értékeit, megszakíthatjuk a programfutást, a töréspontoknál megjeleníthetjük a program által írt képernyőt ...

A debugger segítségével futás közben vizsgálhatjuk programunk működésének helyességét. Akár utasításonként hajtjuk végre a programot, akár az általunk beiktatott töréspontoknál felfüggesztve, tetszőleges változók aktuális értékeit nézhetjük meg, módosíthatjuk is ezeket. A nyomkövetéssel kapcsolatos parancsokat vagy a „Run" menüpontban adjuk meg, vagy az e célra szolgáló funkciós billentyűkkel (menün kívül). A nyomkövetést oktató segédállomány is megtalálható a lemezen. Tíz

Fordító

A fordító opciói:

C (CONTINUE) üzenetek kiadása után nem áll le az Asic, hanem folytatja a fordítást;

L (LIST ALL) fordításkor minden egyes lefordított sort kilistáz (alapértelmezésben csak a hibás sorokat listázza);

P (PRINTLIST) a képernyőn megjelenő üzeneteket ki is nyomtatja;

D (DISkbájtLIST) a képernyőn megjelenő üzeneteket lemezes állományba is tárolja (ez gyorsabb, mint az előző opció);

S (SYMTAB) lemezes állományt készít a szimbólumokról (az összes változó, cimke, belső fordítási címek);

E (EMATH) bekapcsolja az „ Extended Math" opciót (ekkor tudunk long integer típusú változókkal, azaz +2,147,483,647 és —2,147,483,647 közötti értékekkel dolgozni);

X — DEBUG — nyomkövető kódot generál.

A fordító korlátai:

A forrásprogram maximális hossza integrált környezetben 809 sor, önállóan azonban nincs korlátja. A tárgykód maximális hossza 64 kbájt lehet. A forrásprogram egy-egy sora 128 bájt önálló esetben, 80 bájt az integrált környezetben.

Ciklusok 25 szintig skatulyázhatók egymásba. Maga a fordító egymenetes, Borland Turbo C 2.0-ban írták. A forrásutasításokat közvetlenül gépi kóddá alakítja. Teljesen a memóriában dolgozik, így érheti el a legnagyobb sebességet. Lemezterületet egyáltalán nem igényel.

számot rendez „buborék" módszerrel — ezen a példán jól be lehet mutatni a debugger használatát.

Asic-ben nincs EOF, sem ON ERROR. Ezek helyett egy speciális rendszerváltozó (neve ERROR) értékét állítja be akkor, ha állománykezeléskor (OPEN, INPUT#, PRINT#, CLOSE ... utasítások végrehajtásakor) hibát észlelt. Az Asic állományok 2/4 bájtos integer formátumokból állnak (normál/long integer), ellentétben a Basica és GWBasic stringformátumával.

Itt nem olyan szigorúak az írásjelek alkalmazási szabályai. Megengedi ugyan használatukat, de nem követeli meg.

A = ABS (B(1)) helyett nyugodtan írhatjuk:
A = ABS B 1
vagy akár:
A = ABS B 1,,,,,,)))))

A FOR/NEXT ciklusokat legalább egyszer végrehajtja, ez sem így történik Basica/GWBasic-ben. Kerekítéskor (RND) 0 — 32 767 közötti értéket kapunk normál integereknél, 0 — 2.147.483.647 közöttit long integereknél, míg Basica/GWBasic esetén ez 0 és 1 közé esik.

A soros portokat támogató OPENCOM / COMSTAT / SEND / RECEIVE utasítások GWBasic/Basica OPEN/INPUT#/PRINT# -nek felelnek meg. Megengedi a rendszer az idő és dátum lekérdezését, de módosítását már nem.

Új utasítások Asic-ben:

CODE	COMMAND$
INT86	FILEPOS
FIND	LCASE$
LTRIM$	MOD
RTRIM	UCASE$
ZBIT	ZMODE

Ezeknek nincs megfelelőjük sem GWBasic, sem Basica nyelven.

A 124 oldalas kinyomtatható angol nyelvű kézikönyv mellett két Asic nyelven írt demoprogramot is találunk. Az egyik az állományok kezelésére tartalmaz példákat, a másik pedig a valószínűleg közismert cellakeletkezés/-elhalás folyamatának forráskódja. Itt 20x80 karakteres területen minden olyan cella elhal, amelynek kettőnél kevesebb szomszédja van (elszigeteltség), vagy háromnál több (túlnépesedés). Azok maradnak életben, amelyeknek 2 vagy 3 szomszédjuk él. Az üres helyeken új cellák keletkeznek, amikor 3 élő szomszédjuk lesz.

Sándor András

Nyomkövetési tippek

1. Amikor már helyesnek tartjuk programunk működését, tehát nincs tovább szükségünk nyomkövetésre, fordítsuk le újra a DEBUGGING CODE opció nélkül (OFF)! Így nem növeljük feleslegesen sem a futás idejét, sem a kód méretét, mivel gyorsabb és rövidebb programokkal dolgozunk.
2. Figyeljünk arra, hogy míg az integrált környezetben a DEBUGGING CODE ON az alapértelmezés, külön fordítva OFF.
3. Állományok megnyitásakor és fordításakor az addig beiktatott felfüggesztési pontok automatikusan törlődnek.
4. Ha futó programban változtatunk meg egy forrássort (újrafordítás nélkül), akkor a forrásutasítás nem lesz azonos a végrehajtott utasítással. Ezt ne csináljuk soha! Szakítsuk meg a programfutást, és fordítsuk újra!
5. A felfüggesztési pontok helyét a rendszer az editorpuffer elejétől számítja. Kavarodást okozhat a sortörlés/beszúrás, mert a rendszer továbbra is ezzel a relatív sorszámmal dolgozik.
6. Az Asic az INT 3 interruptvektort használja nyomkövetésre. Ennek eredeti értékét akkor állítja vissza, amikor kilépük az Asic-ből.
7. Forrásszintű nyomkövetéshez a DEBUGGING CODE opciót meg kell adnunk fordításkor.
8. Ha futó programunk futását elölről akarjuk kezdeni, először válasszuk a Run menü „Abort your program" opcióját, utána pedig a „Run your program" opciót!
9. „Abort" opció választásakor a programhoz tartozó adatállományok lezárása nem lesz korrekt. (Az adatpufferek tartalma nem kerül vissza a lemezre.)
10. Array változók módosításakor ügyeljünk az indexhatárok betartására! Ezt ugyanis nem ellenőrzi az Asic.
11. Zavaró lehet, hogy a fordító max. 80 karakteres változónevekkel dolgozik, míg a nyomkövető csak 30 karakteresekkel. Erre is nekünk kell ügyelnünk.
12. FOR — NEXT ciklus FOR sorába tett töréspontnál csak egyszer fog megállni a program. Ciklus végén már csak a FOR-t követő utasításra ugrik vissza!
13. A CODE utasítás sorában nem lehet töréspont. Ez logikus, ugyanis egy gépi kódú utasítás több CODE utasítást is tartalmazhat.

A nyomkövetés korlátai: max. 20 töréspont, egyidejűleg 10 szimbólum aktuális értékét figyelhetjük, az egyes szimbólumok maximum 30-30 karakter hosszúak lehetnek, a forrásutasítások száma legfeljebb 809 lehet, a szimbólumok maximális számát nem a nyomkövető, hanem a fordító korlátozza.

Mi muzsikus lelkek...

A nyári SolarSoft újdonságok között hármat is találunk, amely a számítógépes zene iránt érdeklődők számára kellemes meglepetést jelent: az #516 lemez a rég várt Blaster Master 4.8, az #517 a SoundFX-Blast, az #518 pedig az Audiostar programot tartalmazza, Ezúttal az első kettőt ismertetjük.

Azoknak készült a Blaster Master, akik a maximumot szeretnék kihozni PC-jük hangkártyájából. A SoundBlaster és más hangkártyák egyik legérdekesebb tulajdonsága, hogy képesek a hangokat felvenni és visszajátszani is. A Blaster Master alapértelmezésben SoundBlaster típusú .VOC kiterjesztésű állományokkal dolgozik. A regisztrálatlan shareware verzióval csak 25 másodperces hangállományok kezelhetők, a regisztrált verziónál nincs ilyen korlát. Külső hangállományokat is kezelhetünk a rendszerrel : .WAV, .NTI 8SV és .SND formátumokat. A .WAV állományok Microsoft Multi-Media formátumúak, ezeket többek közt a Pro Audio Spectrum és az ATI Stereo F/X hangkártyák támogatják. Az .SND kiterjesztésű hangállományok csak adatokat tartalmaznak, nincs header információ bennük. Az .NTI állományok olyan Amiga mintákat tartalmaznak, amelyeket a Tetra Compositor kezel. A .VOC típusú állományokat mindhárom említett formátumban tárolhatjuk. A .8SV kiterjesztés az Amiga IFF hangállományok alaptípusa. Fel tudjuk ugyan dolgozni az Amiga IFF hangállományokat, de pillanatnyilag még nem menthetjük el saját mintáinkat ilyen típusú állományba.

Maga a program egy igen gyors EGA/VGA grafikus editor, amelynek segítségével kívánságainknak megfele-

lően alakíthatjuk hangállományainkat. Eltávolíthatjuk például a szükségtelen mintákat az elejéről, végéről; kiválaszthatunk részeket, ezekből új állományt képezhetünk. A kivágási pontok pontos kiválasztásakor nagy segítséget jelent, hogy a teljes minta grafikusan jelenik meg a képenyőn, másodpercenkénti osztásban. A feldolgozáshoz a legtisztább pontok kiválasztása a 0 átmenetek alapján történhet. A képernyő színei funkcióbillentyűkkel (F1 — F5) változtathatók. Visszajátszáskor pulzáló jel mutatja, hogy hol tartunk a dallamban. A digitális feldolgozási folyamatok aktivizálása egérrel történik, az esetek többségében az Alt billentyű és a funkció első betűje is ugyanezt az eredményt adja.

Szerkesztéskor blokkműveleteket is végezhetünk. A kiválasztott blokkot megismételhetjük (maximum tízszer), csöndbájtokkal tölthetjük fel, megfordíthatjuk — így visszafelé fogja lejátszani —, visszhangosíthatjuk.

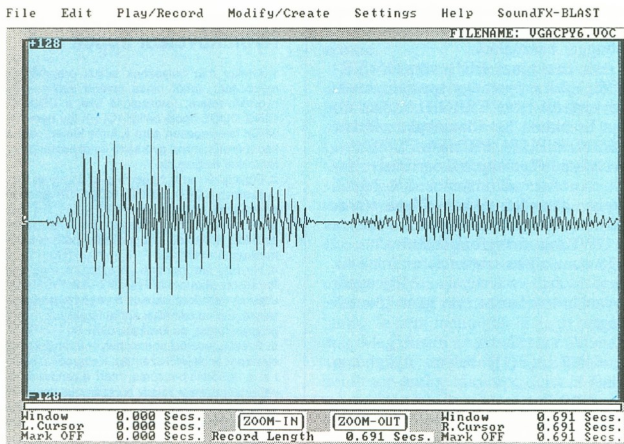
Érdemes kipróbálnunk a 12 000 Hz-en felvett .VOC állományok 8000 Hz-es tárolását. Sokkal jobb hangzást érünk el így, mintha már eredetileg is az alacsonyabb frekvencián vettük volna fel. Úgy lesznek rövidebbek az állományok, hogy a hangzásban nem érződik a tömörítés miatti veszteség. A hangzás tisztasága megmarad akkor is, ha a MAC-en eredetileg 22 000 Hz-cel felvett mintákat 12 000 Hz-en játsszuk le.

Felvételkor érdemes a kisebb hangerőt választanunk, és csak később átállítani nagyobb hangerőre. A SoundBlasternek ugyanis van egy olyan tulajdonsága, hogy nagyobb hangerő esetén erősebb a zavarás. Az említett módszer elsősorban felvételnél ajánlott, ha CD-lemezről vagy magnóról dolgozunk.

Speciális effektusokat érhetünk el két állomány keverésekor a hangerősségek változtatásával.

A Blaster Master hangkártya nélkül is működik, de természetesen ekkor a PLAY, RECORD és a SCOPE műveletek értelmetlenek.

Szükséges konfiguráció: EGA/VGA grafikus kártya, egér, merevlemez, lehetőleg gyors PC. Mivel szerzője lebegőpontos műveleteket használt a nagyobb pontosság elérése céljából, ajánlatosnak tartja, hogy matematikai koprocesszorral is rendelkezzünk. Ez ugyanis a műveletek többségét 40%-kal gyorsítja. Megszívlelendő, hiszen két 50 K-s állomány keverésekor több mint egymillió utasítást hajt végre a rendszer. Lassú grafikus kártya esetén pedig hosszabb várakozási idővel kell számolnunk.



A SoundFX-Blast Editor · grafikus editor SoundBlasterhez. A Silicon Shack cég számos hasonló programot fejlesztett ki: SoundFX, SuperSound, SoundFX-Jr, SoundFX-Voc, SoundCard, SoundBytes, SoundJr és TurboSound.

A SoundFX-Blast segítségével felvehetünk, létrehozhatunk, szerkeszthetünk, módosíthatunk, lejátszhatunk és tárolhatunk hangokat. Semmiféle korlátozás nincs ezzel kapcsolatban: zenét, emberi beszédet, más hasonló természetes hangzásokat vehetünk fel, kezelhetünk. A minőség és az eredeti felvételhez való hűség megtartása már csak rajtunk áll. A rendszer egérrel vagy billentyűzetről vezérelhető, de a fejlesztők egérre optimalizálták — ajánlatos inkább ezt választani.

A rendszer menüvezérelt, párbeszédablakokkal. Az főmenüben File, Edit, Play/Record, Modify/Create, Settings, Help menüpontok jelennek meg. Az egyes pontokból aktivizálható műveletek :

File menü — az aktuális hang hullámforma elmentése állományba akár a jelenlegi néven, akár új néven; már meglévő vagy új állomány megnyitása; kilépés a rendszerből DOS-ba.

Play/Record menü — aktuális hangállomány egyszeri lejátszása elejétől végéig; kijelölt blokk lejátszása; felvett állomány mintavételezési frekvenciájának módosítása; hangerősség beállítása; aktuális teljes hangállomány rögzítése; blokk (hullám kijelölt része) felvétele.

Edit menü — blokk nullázása, törlése; kivágott blokk átvitele/másolása átmeneti pufferbe; átmeneti puffer tartalmának beillesztése a hangminta kijelölt helyére; átmeneti puffer tartalmának hozzáfűzése a hangmintához; átmeneti puffer tartalmának keverése az aktuális hangállomány tartalmával, a kijelölt ponttól kezdődően; csatornaváltás.

Modify/Create menü — már létező hangállomány tartalmának módosítása; hullámfrekvencia korlátozása, módosítása szűrővel; visszhangosítás; visszhangok közti időeltolódás beállítása; amplitúdó módosítás (0 ± 40 dB); a hullámforma hangerejének vagy amplitúdójának nemlineáris illesztése; hangállomány megfordítása (mintha fordítva beszélnénk).

A hangok létrehozása a generátorműveletekkel történik. Zajokat, egyszerű dallamokat, összetett hangzásokat, FM-hangokat tudunk így kialakítani. Generáláskor az aktív hangállományt írjuk felül a hullámforma bal oldalán kijelölt ponttól kezdődően a jobb oldali kijelölésig.

Zajgenerátor — a hullámforma kijelölt részét véletlenszerű fehér vagy nagy sávszélességű zajokkal írja felül. Az amplitúdót mi állítjuk be. A „Filter" művelettel más típusú zajokat is előállíthatunk.

Tiszta szinuszhullámokat , négyszög és háromszög jeleket hoz létre a hanggenerátor, 1 Hz és 20 KHz közötti frekvenciával, beállítható amplitúdóval, az ismétlődési ciklus 1 és 99% között lehet.

Az AM-generátorral amplitúdómodulált szinuszhullámokat állíthatunk elő, megadhatjuk a vivőfrekvenciát, az amplitúdót, a modulálás százalékát. Az FM-generátorral bonyolultabb hangzá-

sokat alakíthatunk ki a frekvenciamodulált szinuszhullámok felhasználásával. Beállíthatjuk a vivőfrekvenciát, az amplitúdót, az eltérési tényezőt. Így a szirénahangtól a géppisztolykattogásig a legkülönbözőbb hangeffektusok érhetők el.

A fentiek alapján egyértelmű, hogy amatőr műhelyben akár ez a demó is elég drága műszerek helyettesítésére (szinusz-, háromszög-, négyszöggenerátor). Blokkos üzemmódja alapján Wobbler-generátor imitálására is alkalmas. (Különböző frekvenciájú blokkokat tudunk egymás mögé tenni.)

Adott mintavételezési frekvenciájú hangállományt át tudunk konvertálni más mintavételezési frekvenciájúba.

A SoundFX-Blast editort és a digitalizált hangokat saját szórakozásunkra, oktatásra vagy akár üzleti célra is használhatjuk. Dolgozhatunk SoundFX-Blast hangállományokkal, de bármilyen más szoftver által létrehozott hangállománnyal is, feltéve, hogy a 8 bites minták ASCII formátumúak. Szinte korlátlan szoftverlehetőségeket nyitnak meg előttünk a digitális hangok, számítsunk azonban arra, hogy időbe fog telni, amíg megtaláljuk a leghatékonyabb technikát az egyedi programokhoz.

Ne feledkezzünk meg arról, hogy a SoundBlaster kártya ugyan tetszőleges hangok kezelésére alkalmas, de természetes hangoknál használható ki legjobban. Azt értjük természetes hangon, amit kellemes, érthető digitalizált beszédhangként hallunk, nem pedig mesterséges, gépi robothangként.

A lemezen található PLAYSB.EXE program nem része az editornak, DOS szintről indítva önállóan játssza le a SoundFX-Blast hangállományokat. Folyamatossá is tehető ez a lejátszás — tetszőleges billentyű leütésével szakíthatjuk meg az ismétlődő dallamot. Több dallam egymás utáni lejátszása is lehetséges. Az egyes billentyűkhöz hozzárendelhetünk billentyűnként 5 hangállományt, 100 ilyen billentyű 500 dallamot reprezentál — ennyit tudunk tehát lejátszani folyamatosan egyetlen utasítással.

A SoundFX-Blast maximális helyigénye DOS 5.0 esetén 350 K. Ez 27 és 82 másodperc közötti felvételnek felel meg.

Konfigurációs igény: IBM PC, 512K memória, matematikai koprocesszor ajánlott, CGA, EGA, MCG, VGA vagy HGA , MS-DOS 2.10 vagy magasabb verzió, SoundBlaster kártya, mikrofon, kábel. A SoundFX-Blast program SoundBlaster kártya nélkül is működik, csak nincs igazán érteleme — nem tudunk sem felvenni, sem pedig lejátszani.

Verebély Pálné

Writer's Heaven

Gépírók a mennyben

A SolarSoft #277 lemeze, a Writer's Heaven „...egy olyan billentyűzetdefiniáló program, amely a népszerű PC-Write szövegszerkesztőbe illesztve, azt napjaink leggyorsabb és legkönnyebben kezelhető szövegszerkesztőjévé teszi" — írja Mark Shepard, a program szerzője.

Bár a legtöbb szövegszerkesztőt a titkárságok részére készítik, ritkán gondolnak az ott dolgozó gépírókra. A gépíró munkája közben ugyanis többször kénytelen a gépelt szövegről a funkcióbillentyűkre tekinteni, hogy az egyes parancsokat kiadhassa. Ez a „kapkodás" lelassítja, elkedvetleníti, és idegessé teszi. Íme a megoldás: a Writer's Heaven, melyet egy gépíró írt, gépíróknak!

A PC-Write külföldön és itthon is nagyon kedvelt shareware szövegszerkesztő, parancskiadásai azonban nem a gépírók kényelmét szolgálják, ezért gondolta úgy a Writer's Heaven készítője, hogy a PC-Write makrózási lehetőségeit kihasználva egy írógépbillentyűzet-központú parancsstruktúrát alakít ki. Nincs egy olyan parancs sem a programban, amelynek kiadásához a gépírónak el kellene mozdítania a kezét a gépírási pozícióból. Az eredmény: egy gyorsan, hatékonyan, egyszerűen kezelhető és könnyen megtanulható szövegszerkesztő — egy olyan editor, amely lényegesen lecsökkenti a szerkesztés idejét. (Ellenben csak tanult gépíróknak ajánlható.)

A Writer's Heaven is shareware. Ez azt jelenti, hogy mindenki ingyenesen lemásolhatja vagy szabadon forgalmazhatja másoknak — de csak változatlan állapotban, és az összes hozzá tartozó állománnyal együtt. A PC-Write 2.6-os vagy annál későbbi verzióival működik együtt. (A PC-Write 3.03 is megtalálható a SolarSoft Programkönyvtárban.) IBM PC vagy azzal kompatibilis gépeken fut. A billentyűzet „A" betűjétől balra található CTRL, ALT és SHIFT gombokat használja, amelyek az IBM gépeken alapkiépítésűek (más kiépítésű billentyűzeteken nem ajánlott).

A makróprogram tulajdonképpen egy .DEF(iniáló) fájl, amely a szerkesztés megkezdése előtt olvasódik be, a billentyűzet újrakiosztását és más utasításokat tartalmaz. A PC-Write kontrollfájljaihoz hasonlóan átírható. Hozzáadja magát a PC-Write parancskészletéhez, így a legtöbb PC-Write funkcióbillentyűt egy másikkal helyettesíti az írógép-billentyűzeten. Teljesen eltávolítja a PC-Write WordStar parancsait, amelyek ott voltak definiálva. A font- és speciáliskarakter-parancsok — az ALT+betű kombinációk — más speciális billentyűkre vannak áthelyezve. Az ENTER/RETURN billentyű SHIFT-STATUS funkciója is módosult. A program monokróm vagy egyszínű CTR monitorral is használható: a betűk sötétek, a háttér pedig világos. A fordított videomód kevésbé veszi igénybe a szemet, mint a normális kijelzés. (Ez a rész minden további nélkül eltávolítható a programból.)

A használat megkönnyítésére a Writer's Heaven tartalmaz egy help képernyőt, amely beilleszthető a PC-Write helpjébe: az első helpképernyőhöz (Basic editing) hozzáíródik a HVN.HLP fájlban található szöveg. Figyelem! Jelenleg a PC-Write automatikus tördelés opciója nem mindig fér össze a Writer's Heavennel. Lassúbb vagy rosszul értelmezett parancsokat, illetve működésképtelen makrókat

eredményez; annál több hibát, minél hosszabb a bevitt szöveg.

De akik lemondanak az automatikus tördelésről, nagyobb sebességgel tudnak majd dolgozni, mert a Writer's Heaven tartalmaz egy tördelési parancsot a sokkal könnyebben elérhető írógépbillentyűk egyikén.

A PC-Write, mint a legtöbb program, azért nem tökéletes, és akadnak benne felderítetlen részek. S valószínű, hogy a Writer's Heavennél is lehetnek olyan problémák, amelyekre nem ad választ a lemezen található részletes leírás. (A klaviatúra változását a lemezmellékletünkön elhelyezett klavia.txt fájl illusztrálja.)

Lampert Csilla

„Komplexitás"

A CPLX program a Turbo Pascal egyik hiányosságát próbálja meg pótolni: alkalmas két tetszés szerinti komplex szám összegének, különbségének, szorzatának, hányadosának képzésére, valamint egy komplex szám n-edik hatványának, s 1—20-adik gyökének számítására. Ezekre a lehetőségekre a műszaki élet több területén is szükség lehet.

A komplex számokat kétféle alakban adhatjuk be, illetve láthatjuk a képernyőn. A felső sorban a három szám (Z1,Z2,Z3) a Descartes-féle koordináta-rendszerben látható, míg alul az Euler-formula szerint. Az ablakokat a <SPACE> billentyűvel, illetve az egérrel válthatjuk.A harmadik ablak (Z3) az eredmények részére van fenntartva, ezért azt nem editálhatjuk, ezenkívül a hatványozás s a gyökvonás műveletét sem alkalmazhatjuk rajta (ebben az esetben a program automatikusan az első ablakra vált). A Z1, Z2 számok értékét EDIT üzemmódban adhatjuk be. Először válasszuk ki a megfelelő ablakot, ezután lépjünk be az EDIT módba! Látni fogjuk, hogy ebben az üzemmódban nemcsak változtatni tudjuk a számok értékét, hanem ki is tudjuk cserélni őket egymással (CHG). Miután újra kiválasztottuk az EDIT pontot (E), beadhatjuk a számok kívánt értékét. Használhatjuk a számbillentyűket, valamint az E billentyűt, s a mínuszjelet. Miután beadtuk a szám egyik komponensét, az ENTER leütése után beadhatjuk a másik komponensét is. Amennyiben a szám megtölti a rendelkezésre álló helyet, a program automatikusan a következő komponensre ugrik.

A műveleteket a megfelelő műveleti jel beadásával aktivizálhatjuk. A műveletet csak az = jel, illetve az ENTER beadása után hajtja végre a gép, ekkor az eredmény megjelenik a harmadik ablakban (Z3). Hatványozás és gyökvonás esetén a program várja a kitevő értékét (x=). Gyökvonáskor x maximum 20 lehet! A műveletet mindig az aktuális számon végzi el a program (kivétel a 3. ablak). Hatványozáskor az eredményt Z3 mutatja, míg gyökvonáskor egy külön képernyőn láthatjuk a gyökök értékeit egymás alatt.

A program bármelyik üzemmódjából az ESC billentyűvel, illetve az egér jobb gombjával tudunk alapállapotba visszatérni. Az üzemmódokat, műveleteket, ablakokat az egérrel is ki tudjuk választani, s a bal gombbal aktivizálni tudjuk. A programból a QUIT módban tudunk kilépni.

A grafika VGA/EGA monitorra készült.

Záruba Károly

IBM PC

SOLARSOFT KATALÓGUS

Lemezszám: 500

Név: C++ Toolkits #3

Szerző: Többek, 1989, 1990.

Leírás: OOP-kiegészítések
A lemezen 20 tömörített állományt találunk:
BAM — (BAM — Bidirectional Associative Memory) Zortech C++ 2.0 nyelven, OOP technikával implementálja a BAM-koncepciót, amelyet Bart Kosko fejlesztett ki. Patrick Simpson bővítéseit is figyelembe vették. BGICLA — (BGI — Borland's graphics Interface) Turbo C++ grafikus rajzoló programok, forrással. Tanulságos!
BROWSE és BROWSE2 — C++ böngészők forráskóddal.
BRENT — Hash algoritmus C++-ban, forráskóddal. R. P. Brent rendezési ötletének megvalósítása mind pszeudokódban, mind C++ nyelven. Brent technikája azon a felismerésen alapul, hogy egyes alkalmazásoknál (ilyenek például a fordítók) a sikeres keresések száma messze meghaladja a beillesztések (insert) számát. Az ő módszere több időt szán a beillesztésekre, mert így esetleg kedvezőbb helyzetek állnak elő.
CCLASS — Osztályöröklés C++-ban, forrással.
COLLEC — Gyűjtő osztály és mintaprogram, amely összeszámolja a kulcsszavakat és az azonosítókat egy forrásfájlban. Jó példa a többszörös öröklésre, bemutatja a tagokra mutató pointerek használatát.
LITERA — AWK alapú preprocesszor, amely nagyban lecsökkentheti a C++ kód dokumentálási nehézségeit. Szöveg és források.
LLIST — „Linked List" osztály C++-ban. Megmutatja, hogyan kezdjünk építeni egy tároló osztályt C++-ban. (Három forráslista)
MENU — Menüző rendszer teljes C++ forrással (6 program).
MOD2CP — A Modula-2 és a C++ összevetése, különös tekintettel a komplex típusokra, 5 forrásprogrammal és a tesztprogram forráslistájával.
NDEL — Az utasítássorban megadott fájlokon kívül mindent töröl. A törlendő állományoknak ugyanabban a könyvtárban kell lenniük, ahol az NDEL program található. „*" használata tiltott. (Forráskód és maga a futtatható program.)
PARSE — LARL(1) nyelvtan C++-hoz. Figyelemre méltó a konstruktor/függvény definíciós probléma megoldása. Több forrásprogram.
PARSER — Gyakorlatilag ugyanazokat a forrásokat tartalmazza, mint a PARSE, csak egyetlen állományban.
RTRACE — Sugárkövetés C++ nyelven, forráskóddal.
STARTUP — A Turbo C++-hoz rendszer-startup kód. Figyelem! Kicsomagoláskor feltétlenül adjuk meg a „-d" opciót, máskülönben egymás hegyére-hátára ír mindent!
TASK — Multitaszking C++-ban. 6 forrásprogram — taszkok közötti üzenetátadás, szemafor osztály kezelése, lock megszüntetése ...
A tesztprogram forráskódban és végrehajtható formában is megtalálható.
TAWK — AWK interpreter erősen kommentezett C++ forrással, főleg a különböző típusú ASCII állományok kezelésére, nyomtatására, 11 forrásprogrammal.
TXTSCR — Egyszerű kiértékelő rendszer C++-ban, komplett forrással.
ZELLER — Zeller 1887-os módszerét kódolták C++-ba. Annak megállapítása, hogy egy naptári hét hányadik napja egy adott nap, igen egyszerű lenne a C++ mktime( ) függvény felhasználásával. Így ennek a forráskódnak inkább elméleti a jelentősége.

Lemezszám: 501

Név: C++ Toolkits #4

Szerző: Többek, 1989, 1990.

Leírás: OOP-kiegészítések
Példa arra, hogyan alkalmazzuk a BISON és a FLEX csomagot saját nyelv készítésére C++, illetve C kódot használva.
A BISON és a FLEX csomagok külön lemezeken vannak, nehogy az állományok összekeveredjenek.
Ez az #501 lemez a BISON-t tartalmazza, a FLEX pedig a SolarSoft #502 lemezen található.
A COM.ZIP tartalmával helyettesíthetjük a hiányzó FLEX/BISON állományokat.

Lemezszám: 502

Név: C++ Toolkits #5

Szerző: Többek, 1989, 1990.

Leírás: OOP-kiegészítések
Az #501 lemezzel együtt használható!

A lemezen 6 tömörített és két szöveges állomány található.
BROWSE — Az aktuális könyvtár *.c, *.cpp és *.hpp állományait végignézve felsorolja a használatban lévő objektumokat.
DATECL — Egyszerű C++ osztály dátumtípushoz. Összeadáshoz, kivonáshoz, dátumok beolvasásához és nyomtatásához szükséges függvényeket tartalmaz.
FLEX — LEX bővítés. Az #501 lemezen található COM.ZIP és BISON.ZIP állományokkal együtt használjuk. Saját fordítót írhatunk vele. A nyelvi kifejezések nyelvtani szabályait és a rutinokat átalakítja C nyelvű függvényekké és táblázatokká. Stringeket olvas be, nyelvi elemekké alakítja ezeket.
OVLOAD — Függvények betöltésével kapcsolatos tesztkészlet. (14 forrásprogram). A függvények adatkonverziót, hibakezelést végeznek.
PARSER — C++ böngésző forrással.
PCTECH — A #340 lemezen található HyperHelper hipertexthelpolvasó egy friss verziója (2.01a) található ebben a tömörített állományban. Ezenkívül a PC TECHNIQUES magazin tartalmából kedvcsinálónak: 10 C (közte 2 C++) forrásprogram, 6 Pascal 2 asszembler (ellenőrző összeget készítenek állományokhoz) és egy Basic forrásprogram.
A két szöveges állomány:
CPLUSS.ASC — Scott Robert Ladd String osztályai C++-ban. Különböző megközelítésekben tárgyalja a C++ sztringeket.
OORULE — 50 hasznos tipp a C-programozóknak, ezek alapján könnyebb lesz az OOP-technika elsajátítása a C++ nyelvet használva.

Lemezszám: **503**

Név: **Turbo C Toolkit**

Szerző: Többek

Leírás: Hasznos segédprogramok — 10 tömörített állomány, két tömörítetlen C nyelvű forrás, egy .EXE állomány.
CSOURC 1.5 — C nyelvű forrásprogramok szervezése, keresztreferencia készítésével. A kiválasztott programok változóit, függvényeit, azonosítóit sorolja fel, kivághatunk programrészleteket, az outputok képernyőre, nyomtatóra, lemezes állományba irányíthatók. Menüvezérelt, jól dokumentált, de a forrást nem mellékelték; szerzője Randy Minder, USA.
DBF — Turbo C 1.0 rutinok dBase III kapcsolathoz. 12 forrásnyelvű rutin, amelyek állományokat nyitnak, zárnak, új állományokat hoznak létre, rekordokat kezelnek. Nem foglalkozik indexállományokkal, adatpufferolással, memomezőkkel. Jól dokumentált, a szükséges könyvtárt is tartalmazza. Szerzője Mark Sadler, USA.
DIRSEL — Függvény, forráskóddal. Megadott állománynév keresése után egy mutatót ad vissza. Szerzője Vernon E. Davis, USA.
DISKLI — Lemezkatalogizáló program forráskóddal. Merevlemez vagy floppy könyvtárait, állományait katalogizálja, visszakérdezhetjük állománysorrendben vagy lemezegységenként. Forráskódot is találunk, de ez csak tanulmányozásra alkalmas, a szerző nem mellékelte saját ablakait, sem a képernyőkezelő rutinok forrását. Csak CGA és EGA grafikus kártyával működik. Szerzője Marvin J. Cook, USA.
ECSRC — Nagyon gyors videorutinok, forrással. A 0-7 videomódokat támogatják, minden képernyőre írt output után aktualizálják a kurzorpozíciót. CGA-ra készültek. Szerzőjük Joe Odenwelle, USA.
EXE2CO — Az EXE2BIN megfelelője. Szerzője, Chris Dunford akkor fejlesztette ki, amikor a DOS 3.3 verziótól kezdve az EXE2BIN már nem került az alapcsomagba, hanem külön meg kellett volna vásárolni a Technical Reference részeként. Teljes dokumentáció, forrásszöveggel.
IBMCOM — IBM RS232 (interruptvezérlésű, soros kommunikációs) rutinok COM1 és COM2 portra. A shareware verzió a small modell könyvtárát tartalmazza. Az input puffer 2000 karakter/port kapacitású lehet, a putsi_com() rutin output puffere pedig 80 karakteres. A sebességet 300 és 112 000 baud között állíthatjuk. Jól dokumentált, forráskódot nem tartalmaz. Fejlesztője az Intrinsic Computer, USA.
INDEX — ISAM fájlkezelő, Turbo C 1.0-ban. Lehetővé teszi, hogy indexállományokat kezeljünk C nyelvű programokból. Egykulcsú állományokat kezel, a kulcs típusa azonban tetszőleges lehet: lebegőpontos, dupla pontosságú, vagy akár a felhasználó által megadott adattípus. Forráskóddal (10 program), könyvtárral, dokumentációval. Szerzője Jim Mischel, USA.
SCRLIB — Turbo C-ben gyors képernyővezérlést tesz lehetővé. Az ANSI.SYS-t gyors kurzorvezérlő rutinokkal tette hatékonyabbá a szerző, Paul Raulerson. Ezeket egy könyvtárba gyűjtötte össze. Használatához az ANSI.SYS-t installálnunk kell a CONFIG.SYS állományban.
TCDEBU — Debugger Turbo C nyelvű forrásprogramokhoz. Jól dokumentált, de sajnos a forrást nem tartalmazza. Szerzője L. David Baldwin.
A két tömörítetlen program:
RMTREE — Teljes alkönyvtár eltávolítása. Rejtett és rendszerállományokra rákérdez.
CCC — Nyomtatóprogram, amely a C kódot blokkosítva nyomtatja ki a jobb olvashatóság kedvéért. (Forráskód nélkül.)
NUMOFF — Rövid forráslista a NumLock gomb kikapcsolására.

Lemezszám: **504**

Név: **PC-Type II 2.0**

Szerző: Richard W. Earl, Buttonware Inc., USA

Leírás: Szuper szövegszerkesztő
A PC-TYPE II 2.0 nagy kapacitású, nagytudású szövegszerkesztő három lemezen. Kibontva 2 Mbájtot foglal el a merevlemezen.
A 29 fejezetből álló dokumentáció 741 kbájt.
Főbb szolgáltatásai a teljesség igénye nélkül:
— Körlevelet készít. Különböző állományok részeit vághatjuk össze az INCLUDE és IF utasítások segítségével.
— Szótár 100 000 angol szóval, a bejegyzések módosíthatók, lecserélhetők, új szavakat vihetünk be. A helyesírás ellenőrzésekor a hibás szavak helyett helyes hasonlót ajánl fel, észreveszi a véletlen szóismétléseket. Keresztrejtvényfejtést segítő programot is tartalmaz.
— Számos, különböző típusú grafikont készíthetünk a táblázatba foglalt adatokból.
— Hexa számokkal is végez műveleteket.
— Vonalakat és kereteket is rajzol.
— Sorszámozás és stringtöbbszörözés, beillesztés adott sorba.
— Visszahozhatjuk a törölt vagy rosszul módosított sorokat.
— Adatbázisok és táblázatkezelők adatai bevehetők, főleg a cég saját fejlesztésű szoftvereiből: PC-File+, PC-File:dB és PC-Calc+. (Soros állományú ASCII adatbázisok és vesszővel elválasztott állo-



Jelenleg mintegy 1000 szoftverből, illetve külföldi szakkönyvből válogathat. Az aktualizált lista megtalálható mostani számunk lemezmellékletén. A megrendelt szoftvert vagy külföldi szakkönyvet postai utánvétellel 2 héten belül házhoz szállítjuk.

MEGRENDELÉS

Megrendelem postai utánvétellel az alábbi termékeket. A vételárat és a postaköltséget átvételekor kifizetem.

A) SZOFTVEREK:

....

....

....

B) SZAKKÖNYVEK:

....

....

....

Dátum:

(aláírás)



Amíg a készlet tart...

Megrendelem az Alaplap kiadványsorozataiban (Könyvek, Füzetek, Lemezek) eddig megjelent (és még kapható) művek közül az alábbiakat:

ALAPLAP KÖNYVEK

... pld: Nagy Gábor: Tömör gyönyör, 256,– Ft

... pld: Kis János–Szegedi Imre: Új víruslélektan, 256,– Ft

... pld: Kis János–Szegedi Imre: Vírushatározó, 256,– Ft

ALAPLAP FÜZETEK

... pld: Detrik Péter: Az SQL nyelvről, 375,– Ft

ALAPLAP LEMEZEK

... pld: Bliss főkönyvi program demója és leírása, 750,– Ft

... pld: Norton Guide keretprogram leírása, 500,– Ft

... pld: PathMinder segédprogram leírása, 1000,– Ft

... pld: CSProlog nyelv leírása, 1000,– Ft

... pld: LIM EMS 4.0 memóriakezelő leírása, 1000,– Ft

... pld: Windows 3.0-hoz magyar betűkészletek, 1000,– Ft





PC Turbo Klub

VÁLASZLAP

Eddigi PC Turbo Klub-tagoknak nem kell kitölteniük és beküldeniük, ha címüket az új címadatbázisban is szerepeltetni akarják. (Lásd erről részletesebben cikkünket az 54. oldalon.)

☐ A PC Turbo Klub tagja vagyok, de kérem, hogy az új címadatbázisban címem már ne szerepeljen.

☐ Nem vagyok a PC Turbo Klub tagja, de szeretném, ha túloldalt megadott címemet az új címadatbázisba felvennék.

Tudomásul veszem, hogy a PC Turbo Klub új címadatbázisába való felvétel esetén címemet az Alaplaptól szakmai címanyagot kérő cégek és intézmények direct mail akcióihoz (meghívók, prospektusok stb. kiküldéséhez) felhasználhatják.

Dátum:

....

aláírás

INFORMÁCIÓKÉRÉS

Kérem, hogy az itt általam BEKARIKÁZOTT KÓDSZÁMÚ hirdetésekkel kapcsolatban küldjenek részemre bővebb tájékoztatást.

Beküldhető: 1992. szeptember 30-ig

ALAPLAP 1992/8 AUGUSZTUS

01	02	03	04	05
06	07	08	09	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50
51	52	53	54	55
56	57	58	59	60
61	62	63	64	65
66	67	68	69	70
71	72	73	74	75
76	77	78	79	80

FELADÓ:

A) Egyéni érdeklődő:

Név: ....

Utca, házszám: ....

Helység: ....

Irányítószám: ....

B) Vállalati érdeklődő:

Cégnév: ....

Ügyintéző: ....

Utca, házszám: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....



Cédrus Kiadó
Pf. 74

Budapest

1441





FELADÓ:

Név: ....

Cég: ....

Utca, házszám: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....



Cédrus Kiadó
Pf. 74

Budapest

1441





FELADÓ:

Név: ....

....

Foglalkozás/Beosztás: ....

Cím: ....

....

Helység: ....

Irányítószám: ....



Cédrus Kiadó
Pf. 74

Budapest

1441



FELADÓ:

Név: ....

Cég: ....

Utca, házszám: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....



Cédrus Kiadó
Pf. 74

Budapest

1441



A LEMEZMELLÉKLET TARTALMA:

- „Biliárdozó marslakók” — avagy a lokális modulok
- Multitask randevú-mechanizmus
- Órajelmegszakítás-számláló
- Szóhosszúsági statisztika-készítés Snobollal
- Alapműveletek komplex számokra
- A kínai számneves feladat megoldása
- A Gerenia kézírásfelismerő rendszer demója
- Tili-toli Herculesen (és a forrásprogram)
- Alaplap Posta — szoftverek, szakkönyvek kínálata
- Új Windows alkalmazások és játékok a SolarSoftban



mányok — Wordstar- — is megfelelőek.)
— 10 különböző állománnyal dolgozhatunk 4 egyidejűleg megnyitott ablakban, ugyanannak az állománynak különböző részeit szerkeszthetjük az egyes ablakokban.
— Az egyes állományokba négy különböző helyet jelölhetünk ki ún. könyvjelzővel, ezek áthelyezhetők.
— Makrózás; egyszerre 25 makró lehet aktív. A makrók hívhatnak más makrókat, illetve önmagukat.
— Tabulátor, lapszél, bekezdés többféle is lehet: egyszeres, kétszeres, háromszoros.
— Lapszámozás; megadható (lehet feltételes vagy automatikus).
— MOVE és COPY utasítások vonatkozhatnak sorokra, bekezdésekre, mondatokra, szavakra, sőt szövegblokkokra (állományok között is).
— SEARCH és REPLACE utasításoknál megadhatjuk azokat az oszlopokat is, amelyekre a keresés kiterjedjen.
— ALIGN és SORT utasítások adatoszlopokra vonatkoznak.
— GET és EDIT utasítások vonatkozhatnak más szöveges állományok részeire vagy egészére.
— SAVE — szerkesztés közben is elmenthetjük az állományok egyes részeit, vagy a teljes állományt. Több mentési másolatot is készíthetünk.
— HEADERS / FOOTERS — Módosíthatók — kikapcsolhatók. (Idő, dátum, lapszámozás, páros/páratlan oldalak megadása.)
— PRINT — Állományok kijelölt részleteit is kinyomtathatjuk. 19 ESC szekvenciát definiálhatunk nyomtatónkhoz.
— A PC-LABEL-t is tartalmazza, így gyorsabb a címkefeldolgozás.
— Környezetérzékeny help rendszer.
— Csaknem minden tevékenység menüvezérelt.
— Az utolsó 10 utasítást tárolhatjuk — későbbi ismételt felhasználásra.
— Általában rendelkezésre állnak a DOS-utasítások a PC-Type II-ből, kilépés nélkül.
— Tetszés szerint konfigurálható, teljesen rugalmas.

Lemezszám: **505**

Név: **Galaxy Lite 1.6**

Szerző: Starlite Software, USA

Leírás: Szövegszerkesztő

Az Omniverse a közkedvelt Galaxy 2.43 verziója után a 3.0-t már nem terjesztette shareware-ként. Azokra a felhasználókra tekintettel, akik megszerették a Galaxyt, és igényelnék a további verziókat is, a Starlite Software kifejlesztette Galaxy Lite elnevezésű szövegszerkesztőjét.
Vezérlése történhet menüből vagy a billentyűzeten begépelt gyors utasításokkal. Az első módszert a kezdőknek ajánlják, míg a második a gyakorlottabbak számára kedvezőbb. A rendszer utasításkészletének kialakításakor azt vették figyelembe, hogy miként szokták használni az emberek a szövegszerkesztőket. Ezáltal olyan rendszert sikerült létrehozni, amelynek kezelése fél óra (!) alatt elsajátítható.
70 000 szavas szótár, helyesírásellenőrzés.
Alt és Ctrl billentyűkombinációkkal aktivizálhatók a formázó, kurzorpozicionáló és a undelete (karakter vagy sor) utasítások.
Funkciós billentyűkhöz rendelték a leggyakoribb parancsokat (18).
Nyomtatás íräsképe (betűtípusok, aláhúzás, fejszöveg, lábjegyzet, ...) beállítható.
Egyidejűleg szerkeszthetünk és nyomtathatunk a rendszerrel.
Makrózási lehetőség — jó makrókészlet, saját makrók írása is könnyű.
Az ablakrendszerrel két állomány közti kivágás/beillesztés valósítható meg.
A zoom lehetővé teszi, hogy bármikor teljes képernyőre nagyítsuk ki a kiválasztott ablakot.
Ismeri a WordStar parancsokat, elfogadja, kezeli a WordStar állományokat.

Lemezszám: **506**

Név: **Prolite 1.01**

Szerző: Quicksoft, USA, 1989.

Leírás: A PC-WRITE könnyített változata
A PROLITE vagy PC-Write Lite a közkedvelt PC-WRITE szövegszerkesztő program kisebb kaliberű változata. Azok számára fejlesztették ki, akiknek laptop gépeiken nincs szükségük a PC-WRITE összes szolgáltatására. Programozók, újságírók, fordítók, diákok például semmit nem fognak hiányolni, ha ezt használják PC-WRITE helyett.
A két program adatformátuma és billentyűparancsai azonosak, így akár felváltva szerkeszthetünk egy szöveget a PC-WRITE és a PROLITE segítségével.
A program már 256 kbájt memóriában is fut (a helyesírás-ellenőrző nélkül), szótára bővíthető, saját nyomtatóinkhoz akár egyedi vezérlőket is készíthetünk.

Lemezszám: **507**

Név: **PC-BROWSE & NG_MAKER**

Szerző: Bob Wallace, Quicksoft, USA, 1990. (PC-BROWSE) Steven W. Kurtz, USA, 1990. (NG_MAKER) John C. Gordon, USA, 1988. (NC_SPLIT)

Leírás: Hipertext és fájlböngésző + NG dekompiler
A lemezen két fő program található. A PC-BROWSE egy univerzális szövegkeresővel kombinált, rezidensen is használható hipertext rendszer, az NG_MAKER csomag pedig a Norton Guide formátumú és a PC-BROWSE formátumú adatbázisok tervezéséhez ad segítséget.
A PC-BROWSE az első olyan elterjedtebbnek számító hipertext rendszer, amellyel nemcsak hogy a képernyőre lehet gyorsan kiíratni az adatbázis megfelelő oldalait, hanem a program keresőfunkcióját kiegészítve a megtalált szövegek közül tetszőleges blokkot is ki lehet jelölni, majd azt az éppen futó szövegszerkesztővel szerkesztett szövegbe be lehet másolni, illetve ki is lehet közvetlenül nyomtatni. (A lemezen található egyik mintapélda a Hamlet „Lenni vagy nem lenni" monológja!)
Az NG_MAKER program segítségével szöveges forrásállományainkat készíthetjük elő mind a PC-BROWSE, mind pedig a Norton Guide programok számára.
A lemezen található még az NG_SPLIT program, angol nyelvű leírással és C nyelvű forráskóddal. Ez az egyik legtökéletesebb Norton Guide-visszafejtő program, nemcsak a forrásszövegeket fejti vissza teljesen komplett és szerkeszthető formába, hanem előállítja a Guide újjáépítéséhez szükséges batch és a linker paraméterezését biztosító listafájlt is.

Lemezszám: **508**

Név: **Proqube Lite 1.0**

Szerző: FormalSoft, USA, 1990.

Leírás: A QUBECALC 3D továbbfejlesztett változata
A közkedvelt háromdimenziós táblázatkezelő, a Qubecalc alaposan továbbfejlesztett változata található ezen a lemezen. A program jelen verziója gond nélkül írja-olvassa a Lotus, DBASE és a megfelelő szerkezetű ASCII állományokat. A maximális kockaméret is alaposan megnőtt: most már 512*512*512 cellából álló munkaterülettel gazdálkodhat a program felhasználója, ami messze több az irodai feladatokhoz szükségesnél. A lemezen — természetesen alaposan összetömörítve — több mint 400 kbájt méretű dokumentáció taglalja a program parancsait, menüit, makróit és használatát. (Ez 186 oldalas kézikönyvet jelent.)
A Lotus 1-2-3 szinte minden függvénye itt is megtalálható, ráadásul a munkakocka mind a hat oldaláról szemügyre vehetjük és kezelhetjük három dimenzióban elhelyezkedő adatainkat. Állományok összekapcsolására is van lehetőségünk. Rendezettséget 9 kulcsig kezel.
Makrónyelve könnyen elsajátítható. Egérrel is vezérelhető a rendszer.

Lemezszám: **509**

Név: **ZIPVIEW/LHA2.13/LHAUTO**

Szerző: FAO Productions, Hollandia (ZIPVIEW) Haruyasu Yoshizaki, Japán (LHA) John A. Qualtrough, Q-Ware, USA, 1991. (LHAUTO)

Leírás: Tömörítők
ZIPVIEW — Kellemes, a Norton Commanderhez készített, de önállóan is használható kukkantó és keretprogram kibontáshoz. .ZIP, .LZH, .PAK, .ARC, .ZOO és .DWC kiterjesztésű állományokat kezel. Sajnos a dokumentáció holland, de a program angol nyelven üzen, így teljesen érthető.
LHA 2.13 Eredeti csomagolásban (ellenőrizhető — az LHA LHA213.EXE paranccsal adjuk Yoshi mesterművének legfrissebb változatát).
LHAUTO — segédprogram (a felhasználó számára) egyetlen lépésben készíti el az SFX archívot a forrásállományokból az LHA segítségével.
A lemezen található NCMAIN.EXE állományra kell lecserélni a Norton Commander központi állományát.

Lemezszám: **510**

Név: **ARJ 2.30**

Szerző: Robert Jung, 1992.

Leírás: Tömörítő
A német Computer Persönlich című lap tesztjében ez a nálunk is méltán népszerű program legfrissebb verziója, a 2.30-as került az első helyre. Olyan vetélytársakat előzött meg, mint például a PKZIP és az LHA.
A tömörítők tesztelésekor a tömörítés hatásfokát, sebességét, a dokumentáltságot és a szolgáltatások sokrétűségét vették figyelembe. Az ARJ hatékonysága legjobban az adatbázisok, tömörítetlen grafikus állományok és a nagy terjedelmű dokumentációk tömörítésekor érvényesült. Volt rá példa, hogy 9 Mbájtos adatbázist 800 kbájtra tömörített!
Az ARJ az egyetlen olyan zömítő, amely többkötetes, együttes archiválásra is képes. Ez azt jelenti, hogy floppykra közvetlenül archivál — függetlenül attól, hogy hány darab és mekkora inputállománnyal kell dolgoznia. Még további lemezterületet sem igényel. Ez a tulajdonság teszi alkalmassá az ARJ-t (DEARJ-t) arra, hogy nagyobb szoftvercsomagok terjesztésének ideális eszköze legyen.
Nem kell ugyanis arra ügyelnünk, hogy egy-egy floppyra teljes állományok férjenek fel. Amikor szükséges, az ARJ automatikusan feldarabolja az állományokat, kicsomagoláskor pedig összerakja a megfelelő részeket. Ezen szolgáltatása révén válhat az ARJ a „szegény ember" backup-segédprogramjává. Az ARJ-vel készített archív állományok tartalmazzák az elérési útvonalakat, az állományok dátum- és időadatait és a fájlattribútumokat. Indexállományt is létrehozhatunk, amely az egyes kötetekre vonatkozó információt tartalmazza, visszakapja.
Magas szintű megbízhatóságot nyújt az ARJ-archívok adatszerkezete: a fejléc (header) felépítése és a 32 bites CRC. Az archiválási folyamat közben is ellenőrizhetjük a tömörítést. Más hasonló programok kitörlik az inputállományt, mielőtt az archív állomány helyességéről meggyőződhetnénk.
A „security envelop" lehetőséggel letilthatjuk (lock) az archív állományt, így ezt az ARJ már nem tudja többé módosítani.

Lemezszám: **511**

Név: **4EDIT & 4ZIP**

Szerző: Többek

Leírás: Kiegészítések a 4DOS programhoz
A SolarSoft #383 lemezen levő 4DOS programhoz gyűjtöttünk össze néhány igen hasznosnak látszó kiegészítést erre a lemezre, négy tömörített, önkicsomagoló állományba:
4DOSMIC — Egér meghajtók Microsoft és Logitech egértípusokhoz 4DOS-ban. Szerző — Tom Ordelman, European Cybernetics, Hollandia.
4DOSPAK — Megjegyzéseket, leírásokat fűzhetünk az állományokhoz ezzel a segédprogrammal. Szerzője Frans Veldman, Hollandia, 1990.
4UTILS — Két segédprogramot tartalmaz. COPYD szerzője David P. Lemire, 1989. Állományok másolására alkalmazható. A KBFIX2 3.0 verzió rezidens, billentyűzettel kapcsolatos bővítéseket tesz lehetővé. Szerzője Skip Gilbrech, 1986.
4ZIP_ED — négy segédprogram :
— 4EDIT 1.03 — Teljes képernyős editor a 4DOS fájlleírások szerkesztéséhez. Szerzője Iram J. Weinstein, USA, 1990.
— 4ZIP 1.04 és 4UNZIP
A 4ZIP arra szolgál, hogy a 4DOS-nál megszokott kommentjeinket ne veszítsük el, ha a PKZIP/PKUNZIP-pel (SolarSoft #421 lemez) dolgozva tömörítjük ritkábban használt állományainkat. Maga a 4ZIP program nem tömörít, csupán kiegészíti a PKZIP és PKUNZIP már megszokott szolgáltatásait. Szerzője Iram J. Weinstein, USA, 1990.
— XDIR 3.0 — Megjegyzésekkel ellátott könyvtárakat kezelő segédprogram. Szerzője Iram J. Weinstein, USA, 1990.
A 4DOSCOLS szöveges állomány elmagyarázza, hogyan oldhatjuk meg 4DOS-nál a színcserét.

Tárgyszerű víziók

A Borland International Object Vision 2.0 for Windows (továbbiakban OV) programja nem más, mint egy Windows-alapú alkalmazásgenerátor — elsődlegesen adatbázis-kezelő alkalmazások „díszlettervezéséhez és megzenésítéséhez".

Az OV nem igényel programozói kvalitásokat, nem kell hozzá programfejlesztői véna; bárki látványos és korrekt (azaz hibátlan) adatbeviteli, adatkarbantartó és lekérdező képernyőket csinálhat vele dBASE- vagy Paradox-kompatibilis adatbázisokhoz.

Az OV ikonszervezésű, barátságos kezelői felületét kihasználva teletömték a programot sok kényelmi funkcióval. Bár billentyűzetről is rövid úton elérhető bármilyen funkció, egérrel a komplexebb szolgáltatások nagyobb része azonnal aktivizálható a jellegzetes ikonokon keresztül. Néhány szolgáltatás vázlatszerű felsorolása:

— Logikai döntésfa vizuális, grafikus reprezentációja (a dBASE-ből ismert VALID függvénynek megfelelő, akár bonyolult feltételrendszer is megadható).

— Egyszerűbb grafikus elemek automatikus megrajzolása: téglalap, keret, kör, ellipszis, árnyékvetés.

— Különböző figyelemfelkeltő betűtípusok, változatos betűméretek.

— A dBASE vagy Paradox adatbázis az OV-ből is közvetlenül megtervezhető és létrehozható.

— Btrieve és ASCII textállományokat is közvetlenül képes beolvasni és kezelni, Sybase/SQL Serverhez csatlakoztatható.

— A DDE (Dynamic Data Exchange) és OLE (Object Linking and Embedding) szoftvertechnológia folytán az OV képes akár hálózatos üzemmódban is együttműködni bármely dBASE vagy Paradox programmal, azokkal közösen, és egyidejűleg képes adatállományokat karbantartani.

(Egészen extrém esetben elképzelhető a következő is: Windows-os felületen kommunikál programunk a tisztelt felhasználóval, de az adatfeldolgozást egy, a háttérben futó, Clipperben írt program végzi.)

— Az MS Windows révén minden nyomtatót gond nélkül kezel.

— Objektumorientált képernyőtervezés, minden képi elem és adatmező könnyedén elmozgatható, áthelyezhető, mérete, színe, betűtípusa, vonalvastagsága egyszerűen megváltoztatható.

A csomag tartalmazza az Object Vision 2.0 royalty free runtime változatát is, ez az a program, amelyet az OV-ban írott alkalmazásainkhoz mellékelhetünk, mivel ez a verzió nem tartalmazza az interaktív tervezői funkciókat. A programozók fantáziájának megmozgatása érdekében még további, mintegy húsz kész, tetszés szerint módosítható alkalmazást is mellékeltek az OV Runtime lemezén.

Ára az Alaplap Postán keresztül 16 800 Ft + ÁFA.

Harvard Graphics for Windows 1.0

Szabvány teremtője

A klasszikusnak mondható Harvard Graphics az amerikai Software Publishing Corporation terméke (más ismert Windowsos termékük a Harvard Draw és a Superbase 2 és 4). A Harvard Graphics (továbbiakban HG) prezentációs programcsalád, melynek már nemcsak DOS alatt futó, hanem Windowsos implementációja is létezik (ezúttal ezt a változatot ismertetjük behatóbban).

Valódi piacgazdaságokban egy valamirevaló cég menedzserei rendszeresen, évente több alkalommal is bemutatják a vállalkozás részvényeseinek és igazgatótanácsának az adott üzleti időszak addigi eredményeit és a következő üzleti periódus megvalósítandó terveit. A legszemléletesebben diavetítéssel, számítógépes demonstrációkkal színesítve adják el(ő) mondandójukat a nem feltétlenül képzett közgazdászokból álló hallgatóságnak. A lustábbja már csupán számítógépet használ, és abban a hitben ringatja magát, hogy lépést tart a korral.

A HG a korábbi megszokott eszközök (írásvetítő fólia, diakockák, előre megrajzolt, sokszorosított és kiosztott grafikonok stb.) helyett vagy mellett azok hatásán túltesz: képes önálló számítógépes szemléltetésre is, akár végtelenített (avagy interaktív) slide-show-k megtervezésére és megjelenítésére is. Persze egy PC-s demonstráció képeit egyenként is ki lehet nyomtatni, sokszorosítás és bekötés után a prezentáció emlékeztetőjét az előadás kezdete előtt pár perccel ki is lehet osztani, amibe a jelenlévők menet közben bejegyezhetik észrevételeiket, kérdéseiket.

A fent leírt technológia minden egyes fázisát támogatja a HG programcsomag, kezdve a bemutató vázlatának megírásától a feliratozáson keresztül az egyes grafikonok, szemléltető diagramok automatikus elkészítésén át magáig a szabadon futó prezentáció megjelenítéséig képernyőn — illetve mindezeknek kiíratásáig lézernyomtatón, plotteren, lefényképezéséig instant diakészítővel. A program a következő szolgáltatásokat nyújtja:

— vázlatszerkesztő (outliner) és szövegszerkesztő;

— angol nyelvű helyesírás-ellenőrző külön egy saját magunk által is szabadon bővíthető és módosítható kivételszószedettel;

— általános grafikon- és szövegszerkesztő (Chart Gallery): főcímek, alcímek, felsorolások, táblázatok, hierarchikus blokkdiagramok, oszlop-, vonal-, kör-, minimum-maximum, terület-, pont- és szalagdiagramok, vagy az előbbiek tetszés szerinti kombinációja 2 vagy 3 dimenziós effektusokkal, szí-

nekkel, látványos betűtípusokkal feltupírozva (mintegy 80-féle diagramtípus);

— beépített kis táblázatkezelő a grafikonadatok kényelmes és egyszerű beviteléhez (oszlop- és sorfunkciók, átlagszámítási és egyéb statisztikai algoritmusok);

— slide-show megjelenítő, „diarendező üzemmód" (ScreenShow);

— HyperShow: programozható, interaktív ScreenShow, mely alkalmas szabadonfutó vásári demók, menüvezérelt oktatóprogramok és más szabad formátumú prezentációk megkomponálásához (látványos megoldások: áttűnések, képek be- és kiúsztatása, elhalványítása, lepergetése stb.);

— 30 stílusminta, több mint 500 kész sematikus ábra, és rugalmasan változtatható méretű szimbólum (nyilak, keretek, emberkék, országok sziluettje — köztük Magyarország normál és megyetérképe, valamint színes zászlaja —, piktogramok, szabványos folyamatábra-elemek, stb.);

— a legkorszerűbb szoftvertechnológiák felhasználásával (DDE = Dynamic Data Exchange és OLE = Object Linking & Embedding) a HG kétirányúan képes együttműködni más Windows-alkalmazásokkal (pl. MS Excel, 1-2-3 for Windows);

— beépített objektumorientált rajzolóprogram: bármi elforgatható, tükrözhető, torzítható, a méretek fokozatmentesen állíthatók, színeket, kitöltő mintákat, rasztereket használhatunk;

— strukturált, hierarchikus információkezelés, vázlatkészítés támogatása.

A világon a prezentációs szoftverek közül nem véletlenül a Harvard Graphicsból adták el a legtöbbet, jelenleg már közel két és fél millió darabnál tartanak.

Minimális hardver- és szoftverkövetelmények

— Bármely PC, melyen fut az MS Windows 3.x standard vagy enhanced üzemmódban (286, 386 vagy 486-os processzorral), de legalább 386-os az ajánlott.
— DOS 3.1 vagy efölötti verzió.
— EGA vagy VGA, utóbbi ajánlott.
— Egér nyomatékosan ajánlott, de nem szükséges, tablet (mutatótábla) pótolhatja az egeret.
— Minimum 2 Mbájt RAM, de 4 Mbájt az igazi.
— Minimum 8 Mbájt harddiszkkapacitás, de az összes opció installálásához 14 Mbájt szükségeltetik.

Támogatott outputeszközök és szoftverek:

— Az összes Windowson keresztül vezérelhető printer, plotter és direkt dia- vagy filmfelvevő, többek között az Apple, Agfa, Canon, Epson, Hewlett-Packard, IBM, Lasergraphics, NEC, Polaroid, QMS és Tektronix modellek.
— Videokártyák: EGA, VGA, SuperVGA, 8514/A, amelyek legalább 256 kB RAM-mal rendelkeznek.
— Hálózatok: 3Com 3+ Share, Novell ELS, Advanced, SFT és Netware 386, IBM PC LAN, Banyan Vines 386, Microsoft OS/2 LAN Manager (DOS Client), IBM OS/2 LAN Server (DOS Requester) — ha több felhasználó is akar egyidejűleg dolgozni, úgy 1, 5 vagy 20 felhasználós ún. LAN License Packokat kell még vásárolni (áruk rendre a teljes értékű csomag árának 0,7, 3 vagy 13-szorosa);
— Fonttechnológiák: Bitstream Speedo (a csomagban benne foglaltatik), Bitstream Facelift, Adobe Type Manager;
— A korábbi DOS-os Harvard Graphics 2.x és 3.0 chart és symbol állományait képes feldolgozni, beolvas Lotus 1-2-3, MS Excel és ASCII textállományokat.

Az Alaplap Posta útján többféle áron érhető el:

Harvard Graphics for Windows	1 053 000+ ÁFA
Harvard Graphics f/W Upgrade	19 000 + ÁFA
Harvard Graphics 3.0 for DOS	53 000 + ÁFA
Harvard GeoGraphics for DOS	39 900 + ÁFA

Demólemez: 299 Ft + ÁFA, melynek árát visszakapja az, aki később a teljes értékű szoftvert is megvásárolja az Alaplap Postától.

Harvard Draw for Windows 1.0

A CorelDRAW méltó vetélytársa

A prezentációs programok ősatyját, a klasszikusnak mondható Harvard Graphicsot követő Harvard Draw for Windows (a továbbiakban HDW) az amerikai Software Publishing Corporation terméke. A HDW-t mint univerzális rajzolóprogramot elsősorban desktop publishing (DTP) programokhoz ajánlják: ezek kiegészítésére, színesítésére ábrákkal; illetve önálló grafikai műalkotások létrehozására szánják.

A félreértések elkerülése végett: a HDW-nek semmi köze a Bostonban található híres Harvard Egyetemhez. Emez Kaliforniában készült.

Tekintettel arra, hogy nagy lemezfaló program, mely teljes installálás esetén 30 Mbájt szabad helyet tenne magáévá, valamit tenni kellett. A megoldás: az intelligens üzembe helyezésű program megengedi, hogy akár fájlszinten mazsolázhassunk a kínálatból. Ez különösen áll a töméntelen mennyiségű grafikus ábrára és szimbólumra. Mi is csak a legfontosabb szolgáltatásait soroljuk.

— A szabadkézi rajzolás messzemenő támogatása: az egyik ablakban kinagyítva dolgozunk a részleteken, míg egy másik ablakban teljes színeiben és nagyságában pompázik művünk.

— Bármikor több színpalettából válogathatunk, sőt saját tervezésű színpalettákat is kifejleszthetünk. A színhatásokat tovább fokozhatjuk különböző kitöltőminták alkalmazásával. Egy adott területet akár befoglaló vonalainak megfelelően körkörösen, csillagszerűen, avagy adott színátmeneteken keresztül tölthetünk fel (elég megadni a kiindulási és a végső színt).

— A HDW objektumorientált program, így minden egyes részletet külön kezel, a teljes ábrát meglehetősen nehéz óvatlan mozdulatokkal elrontani. Ha

mégis mellényúlnánk, a 16 szintű visszaállító funkció (undo) mindig segít.

— A legtöbb hasznos funkció nemcsak menükön és billentyűkombinációkon keresztül érhető el, hanem külön ikontömbök (floating dialog boxes) könnyítik meg a program gördülékeny kezelését.

— Autotrace funkció: egy más rendszerből átvett rajz vagy szkennerrel beolvasott ábra körvonalait automatikusan kiegyenlíti!

— Online helprendszer.

— Ún. Harvard Draw Script Language, mellyel összetett ábrákat generálhatunk egyetlen tollvonás nélkül, normál ASCII-szöveges formában, mérnöki pontossággal!

— Geometriai alapformák rajzolása: kör, ellipszis, négyszög, sokszög, szabályos ívek, szögek. Ábrák tengelyes és középpontos tükrözése, elforgatása, döntése, torzítása, fokozatmentes zoom.

— Használható mértékegységek: inch, centiméter, pont.

— Fóliaszerűen egyszerre 99 átlátszó lapot tud kezelni, ezeken tetszés szerinti grafikus objektum szerepelhet.

— Speciális effektusok: két eltérő alakzatot képes egymásba átvinni a közbenső fokozatok megszerkesztésével, képrészletet megsokszoroz, maszkol (egy képből egy minta szerint vág ki részletet).

— Szövegek megadott vonalra, mintára illesztése, árnyalása, kitöltése, körvonalazása.

— A csomag ráadásképpen egy műanyag vonalzót és betűmércét is tartalmaz!

A HDW relatív gyorsaságában és hatékonyságában tűnik ki a többi hasonszőrű program közül. Egy szórólapon éppen a CorelDRAW 2.0-val hasonlítják össze, a versengésben a HDW győzött, a mérkőzés végeredménye 21:7 lett.

Javarészt sebességben múlta felül vetélytársát, néhol bizony tízszer, tizenkétszer is sebesebben oldva meg ugyanazt a tesztfeladatot. Mint tudjuk, az idő pénz. De ha egy program — amiért mi magunk fizettünk — húzza az időnket, és borzolja idegszálainkat, itt az ideje, hogy lecseréljük egy olcsóbb fürgébbre...

Minimális hardver- és szoftverkövetelmények

— Bármely PC, melyen fut az MS Windows 3.x standard vagy enhanced üzemmódban (286, 386 vagy 486-os processzorral), de legalább 386-os az igazán megfelelő.
— DOS 3.1 vagy efölötti verzió.
— Hercules, EGA vagy VGA monitor; ez utóbbi leginkább, mely képes 256 szín megjelenítésére 640x480-as felbontásban is.
— Egér nyomatékosan ajánlott, de nem szükséges.
— Minimum 1 Mbájt RAM, de 4 Mbájt a legjobb, melyből 2 Mbájt RAMdrive-ot érdemes lefoglalni az átmeneti állományok gyors kezelése érdekében.
— Minimum 10 Mbájt harddiszkkapacitás, de az összes opció és mintaállomány installálásához plusz 20 Mbájt szükségeltetik.
— Ha nagy bitmap ábrák importját is tervezzük, hozzunk létre a Windows SWAPFILE nevű segédprogramjával egy 8 Mbájtos átmeneti állományt, amennyiben 386-os gépünkön enhanced módban kívánjuk futtatni a HDW-t.

Támogatott outputeszközök és szoftverek:

— Az összes, Windowson keresztül vezérelhető printer, plotter és direkt dia- vagy filmfelvevő, többek között az Apple, Agfa, Canon, Epson, Hewlett-Packard, IBM, NEC, Polaroid, QMS és Tektronix modellek.
— Speciális berendezések: General Parametrics Videosow, Agfa Matrix SCODL, Lasergraphics LFR/PFR, PTI Montage FR1.

— Videokártyák: EGA, VGA, SuperVGA, 8514/A, amelyek legalább 256 kbájt RAM-mal rendelkeznek.
— Hálózatok: 3Com 3+ Share, Novell ELS, Advanced, SFT és Netware 386, IBM PC LAN, Banyan Vines 386, Microsoft OS/2 LAN Manager (DOS Client), IBM OS/2 LAN Server 1.3.
— 47 beépített Bitstream skálázható betűcsalád.
— Harvard Graphics 2.x, 3.0 chart és symbol állományait képes beolvasni.
— A beemelhető vektor- és bitmapped állományok: AI, PCX, WMF, DRW, BMP, TIFF, CH3, CHT, CGM, SYM, DXF.
— Kimenő (export) állománytípusok: SCODL, AI, TIFF, PCX, BMP, WMF, CH3, PIC, CGM, SY3, DXF, LL, EPS, Windows clipboard.

Az Alaplap Postán keresztül többféle áron érhető el:

Harvard Draw for Windows 1.0	59 000 + ÁFA
Harvard Draw f/W Competitive Upgrade	19 000 + ÁFA

(Konkurens programok — például CorelDRAW, Micrografx Designer, GEM Artline, Adobe Illustrator, Arts & Letters, PC Paintbrush, Aldus Freehand — felhasználói a program eredeti kézikönyvének első oldalát átadva jogosultak a kedvezményes bevezetési árra.)
Demólemez (1 db 1,2 MB-os 5,25"-es lemez): 299 Ft + ÁFA, melynek árát visszakapja, aki később a teljes értékű szoftvert is megvásárolja az Alaplap Postától.

E számunk hirdetői

Civilizált Ön?

Játéktér: a jó öreg anyaföld

Hazánkban is sokan játsszák szívesen a különböző szimulációs játékokat. Ezek közül is a legizgalmasabbak a nem csupán egy eszközt (vadászrepülőt, harci helikoptert vagy tengeralattjárót), hanem egy komplett önszabályozó rendszert modellező programok.

Már az Alaplap hasábjain is megismerkedhettünk a SimCity és SimEarth nevű zseniális játékokkal. (Megjegyezzük, hogy a Maxis — az előbbi két termék kiagyalója — azóta megjelentette a SimAnt nevű újabb „dilijét" is; itt egy hangyatársadalom tevékeny részesévé avat a program.) Ezúttal szintén egy olyan programot igyekszünk bemutatni játékos kedvű olvasóinknak, mely térben az egész földet (sőt a játék vége felé a világűrt is) felöleli. Időben pedig az értelemmel bíró ember megjelenésétől (Krisztus előtt 4000-től) az első űrkolóniák letelepítéséig fogja át a korokat. A program neve: Civilization!

A Civilizációt a Microprose sikeres szerzője, Sid Meier és csapata készítette. (Korábbi világszámuk az 1990-ben az év stratégiai játéka címet elnyert Railroad Tycoon volt.) A játékos hanyatló birodalmakat virágoztathat fel logikus döntéseivel. Induláskor választható ugyan más geofizikai és klimatikai paraméterekkel rendelkező bolygó is, de higgyék el, sokkal borzongatóbb úgy játszani, hogy az ellenséges népek földrajzilag ugyanazon a helyen találhatók, ahová őket a világtörténelem predesztinálta. Az ókori görögök a Balkánon, a rómaiak az Appennini-félszigeten, a mongolok Belső-Ázsiában, az aztékok a mai Mexikó területén stb...

Nem a szakszervezeti vezető!

A játék egyúttal társadalom-, gazdaság- és technikatörténeti leckeként is felfogható. Több nagy történelmi alakkal is találkozhatunk a szórakoztató időutazás során, így felbukkan Abraham Lincoln, a macedón Nagy Sándor, az angol I. Erzsébet királyné, a mongol világhódító Dzsingisz kán, a babilóniai Hammurabi, a római Julius Caesar, az indiai Mahatma Gandhi, a kínai Mao Ce-tung, az azték Montezuma, a francia nagy Napóleon, az óegyiptomi Ramszesz, a zulu származású Shaka, az orosz (tudjuk: grúz) Sztálin és II. Vilmos Frigyes, aki nem kevesebb, mint ötven évig volt Poroszország uralkodója.

A játék alapegysége a város. Kezdetben 1, 2 vagy 3 vándorló hordával kezdhetjük a nagy civilizációs kalandot. A hordák alkalmas terepviszonyok és földrajzi fekvés, növényzet, folyó vagy állóvíz közelsége esetén letelepedhetnek, és várost alapíthatnak. A városok kezdetben önellátók. Ha van elegendő élelem, a város lakossága folyamatosan gyarapszik. Megszervezik a városi milíciát, később önálló portyázásra is alkalmas bandériumokat, szabadcsapatokat is toboroznak. A városlakók adót fizetnek nekünk; mi a fővárosban (az elsőnek alapított településen) élvezzük a javakat egyre bővülő királyi palotánkban.

Kenyér és cirkusz

Idővel az emberek közül kinevelhetünk csepürágókat, színészeket és egyéb, a nép szórakoztatását szolgáló művészeket. Ezekből később adószedő tiszteket, majd pedig tudósokat emelhetünk ki. Utóbbiak meggyorsítják egy város szellemi és anyagi gyarapodását, míg a színészfélék a lakosság jó kedélyállapotát vannak hivatva szolgálni. Mert bizony, ahogy azt a történelemkönyvekből is megtanulhattuk, az elégedetlen, jóllakatlan ember nem szereti a rendet, lázadozhat, sőt urambocsá', forradalommal le is taszíthat bennünket a trónról.

A városok a technika fejlődésével összhangban egyre ütőképesebb seregeket állíthatnak. Mennél nehezebb és fejlettebb fegyverzettel küldik őket útjukra, annál biztosabb, hogy idegen városokat, népeket hódítanak meg vagy törölnek el a Föld színéről.

A Civilizáció ötféle nehézségi fokozatban játszható: Chieftain (törzsfőnök), Warlord (hadúr), Prince (herceg), King (király) és Emperor (császár). A feladatunk egyre nehezedik, ahogy a magasabb rangú fokozatot választjuk. A legkönnyebb fokozatban még hasznos tanácsokat kapunk főtanácsosainktól. A nehezebb fokozatokban már egyre harciasabbak az ellenséges törzsek, népek, és az újabb technikai vívmányok kifejlesztése is több időt vesz igénybe. A kezdetek kezdetén megadhatjuk, hogy hány törzs boldoguljon velünk együtt a földkerekségen, háromtól hétig szabadon kijelölhetjük riválisaink számát. Majd nemzetiséget is választhatunk, s a választásunknak megfelelően azonnal egy ismert személyiség nevét kapjuk kölcsön.

A játék pontra megy. A pontokat a program bonyolultabb összefüggések alapján számolja, s bizony szűkmarkúan méri. A kapott pontok száma természetesen korrelál az össznépességgel, a társadalom megelégedettségével, a kiépített infrastruktúrával, a hódítások számával, és a kereskedelem és banktechnika minőségével. Bármikor nyugállományba vonulhatunk, ekkor a program kiértékeli addigi munkánkat. Lehetőség van a mindenkori állást el is menteni lemezre, s a korábbi állást bármikor vissza is tölthetjük. A játéknak nyilvánvalóan akkor is automatikusan vége szakad, ha az egyik nép utolsó városunkat is leigázza, azaz kihalunk, illetve ha az összes népet/nemzetet egyenként meghódítottuk.

Tájak, korok, emberek

Az idő rohamléptekkel telik. Kezdetben 20, majd 10 évet lép egyszerre az idő visszafordíthatatlan kereke, később 2, utóbb egy évet, de ekkor már 15-20 várost kell kormányozni. A helyváltoztatás léptéke körülbelül megegyezik Magyarország mostani területével. Az új helyen sokféle növényzet fogadhat bennünket: sarkköri, sivatagi, hegyi, dzsungel, magas hegységi, óceáni, al-

földi, folyómelléki, mocsaras, tundra vagy erdei. A mezőgazdaság mielőbbi kifejlesztésének érdekében természetesen az alföldi és erdei növényzet és talajviszonyok a legkedvezőbbek, de megfelelő öntözési kultúrával (amit ki kell fejlesztenie tudósainknak és feltalálóinknak) még a sivatag is termővé tehető. A következő természeti kincsekből gazdálkodhatunk, ha találunk persze: szén (a hegyekben), hal (az óceánból), vadak (erdős területen vagy tundrán), nemesfémek, drágakövek (dzsungelben), arany (magashegységekben), lovak (síkságon, alföldön), oázis (sivatagban) és kőolaj (mocsárvidéken).

A barbár szín: piros

A nagyobb népektől függetlenül bárhol felbukkanhatnak kisebb harcias, barbár törzsek, amelyeknek egyedüli célja a zsákmányolás és pusztítás. Színük mindig piros. Az ellopott technikai vívmányok nyomán egyre jobb hadifelszereltséggel és kultúrával rendelkeznek, így mind nehezebb lesz őket visszaverni.

Az alapképernyő hat fő részre tagolódik. A legfelső sor a menüléc, innen minden egyes funkció rámutatással is aktivizálható, de a parancsok és beavatkozások zöme egyszerű betűparanccsal vagy egérrel is kivitelezhető. A bal felső nyolcadban található a kicsinyített sematikus világtérkép. Alatta fényes palotánk kicsinyített mása (melynek pompázata egyenes arányban áll sikerességünkkel, a nép általunk érzett megbecsülésével és hódolatával. Ez alatt látható a státusablak, mely állandóan jelzi nemzetünk összlakosságának számát, az aktuális évszámot, egy villanyizzót, amely a tudományos fejlődést van hivatva jelképezni (mennél világosabb, annál közelebb vannak tudományos munkatársaink a legközelebbi korszakalkotó felfedezéshez. Ugyancsak itt jelzi egy kis ábra a környezetszennyezés mértékét is, ez inkább az iparosodás korától válik komollyá. A következő sor a birodalmi kincstár tartalmát mutatja, mellette a gazdaság legfontosabb mutatói, a fényűzésre, az adóra és a tudományra fordított összegek aránya. Ez alatt helyezkedik el az aktuális mozgó egységet jellemző azonosító ablak. Információt ad a kérdéses egység nemzetiségéről, típusáról (katonai, szállító, kereskedő, diplomata, telepes stb...), megtudható a rangja (például veterán, ha már több csatanyerés van a háta mögött), hátralévő lépésszáma, az őt anyagilag finanszírozó város neve, a lefoglalt terület növényzete, valamint látható az ugyanazon a mezőn táborozó más egységek szimbolikus képe (melyek fekete kerettel a megerősített, „beásott" egységek, árnyaltan az őregységnek kirendelt csapatok, amelyeknek nem mozdulnak el a helyükről, csak ha az őket küldő város visszarendeli őket). Az őrök automatikusan teszik a dolgukat, ha ellenséges erő közelíti meg őket.

Egy egységet három fő adat jellemez: a támadó (A = attacking), a védekezési (D = defense) és mozgási (M = movement) faktor. A hadba fogható egységeket a relatíve magas A és D faktor jellemzi.

A tengeren is járhatunk, de csak korszerű tengeri alkalmatosságokkal, úgymint: gálya, flotta, vitorlás, teherhajó, csatahajó. Az iparosodott korokban már tengerről és levegőből is érheti támadás a városokat. A kisebb városok eltűnnek a a térképről, míg a nagyobbak egyszerűen gazdát cserélnek. Mód van seregek feloszlatására (disband) és ember alkotta eszközök (hidak, öntözőrendszerek, bányák) szándékos lerombolására.

A telepesek úttörő szerepet látnak el az egész játék folyamán. Valójában a tudósokon kívül ők képviselik egyedül a produktív szférát. Termővé teszik a földeket, öntözőműveket hoznak létre, bányákat tárnak fel, mocsarakat csapolnak le, erődítményeket és városokat építenek, megszüntetik a környezetszennyezés okozta károkat; utakat, a főbb utak, kereszteződések mentén pedig vasutakat építenek (kellő technikai peremfeltételek esetén).

Lám-lám...!

A diplomaták (mint az életben már megszoktuk) nem éppen tiszta játékot űznek. Egy városba vagy egy ellenséges csapat tagjai közé belopakodva a következő diplomáciai erőfeszítéseket hajthatják végre: egy város erőforrásait kémkedik ki, nagykövetséget alapíthatnak, ipari titkokat deríthetnek föl, azaz ellophatnak idegen technológiát, szabotázs- és terrorakciókat vihetnek véghez (híd-, templomrombolás, a városfalak meggyengítése), forradalmat szíthatnak a köznép körében, találkozhatnak a riválís uralkodóval, végül jó pénzzel meg is vesztegethetnek és rávehetnek átállásra ellenkező oldalon szolgáló egységeket. Békét köthetnek, áruba bocsáthatnak techológiákat, hadat üzenhetnek, szövetségeket köthetnek harmadik nemzet rovására.

A kereskedelmet a karavánok alapozzák meg. A kereskedés hoz be az országba luxustárgyakat, javakat, építőanyagot a világ hét (meg még 14) csodájának megépítéséhez. Bármely honi, 10 mezőnél távolabbi várost elérő karaván létrehozhat ún. országutat, amelyen később vámot szedhet. Idegen kultúrák között teremtheti meg a békés kapcsolatot.

A legváltozatosabb palettát az egyes fegyverek, seregtestek mutatják. A korokban és a korok között gátlástalanul csapongva megpróbáljuk felsorolni a lehetőségeket, feltüntetve a fentebb említett A-D-M faktorokat is:

Armor (páncélozott hadosztály, automobil megléte szükséges):	10-5-2
Artillery (tüzérség, robottechnika kell):	12-2-2
Battleship (hadihajó, acélipar kell):	18-12-4
Bomber (bombázó, fejlett aviatika):	12-1-8
Cannon (ágyú, fémkohászat):	8-1-1
Carrier (repülőgép-anyahajó, fejlett aviatika):	1-12-5
Catapult (katapult, fejlett matematikai ismeretek):	6-1-1
Cavalry (lovasság, lovaglás ismerete):	2-1-1
Chariot (harci szekér, a kerék ismerete):	4-1-2
Cruiser (romboló hadihajó, robbanómotor ismerete):	6-6-6
Fighter (vadászrepülő, repülés ismerete):	3-3-10
Frigate (fregatt, mágnesesség ismerete):	2-2-3
Ironclad (páncélozott monitor, csatahajó, gőzgép ismerete):	4-4-4
Knights (lovagok, a lovagi rendek megjelenése):	4-2-3
Legion (légiók, vasmegmunkálás ismerete):	3-1-1
Mechanized Infantry (gépesített gyalogság, szakszervezetek?!):	6-6-3
Militia (bandérium):	1-1-1
Musketeers (muskétások, puskapor megléte):	2-3-1
Nuclear (atommeghajtású rakéta, rakétatechnika és magfúzió):	99-0-16
Phalanx (falanx, bronz-megmunkálás):	1-2-1
Riflemen (mesterlövészek, zsoldos katonaság intézménye):	3-5-1
Sail (vitorlás hajó, navigáció ismerete):	1-1-3
Submarine (tengeralattjáró, tömeggyártás kifejlődése):	8-2-3
Transport (teherhajó, iparosítás):	0-3-4
Trireme (gályahajó, térképkészítés ismerete):	1-0-3

Az államformák sorrendben a rabszolgatartó társadalomtól az anarchián keresztül monarchia, kommunizmus, köztársaság és demokrácia lehetnek. Érdekes módon a programban pontosan ebben a fejlettségi sorrendben alakulhatnak ki! Elgondolkoztató. A mi államformánk: Magyar Köztársaság. Jó úton haladunk a demokrácia irányába.

Sertepertélő tanácsadók, világcsodák

Hatféle tanácsadó sertepertél körülöttünk. Egy városi (belügyi vagy inkább önkormányzati?!), egy katonai, egy hírszerzési, egy közhangulati, egy kereskedelmi és egy tudományos tanácsadó. Nem árt megfogadni, amit súgnak, azonban gyakran két tanácsadó egymásnak ellentmondó instrukciókat ad, ilyenkor nekünk kell döntenünk, melyiket fogadjuk meg. Különböző érdekes statisztikákat is kérhetünk: a világ csodálatára méltó teljesítmények (Colosseum, nagykönyvtár, kínai nagy fal, függőkertek, világítótorony, jósda, piramisok, Kopernikusz csillagvizsgálója, Darwin utazása a Galápagos-szigetekre, Isaac Newton iskolája, J. S. Bach katedrálisa, Magellán expedíciója, Michelangelo kápolnája, Shakespeare színháza, az Apollo űrprogram, a rák ellenszere, Hoover Dam — az első vízi erőmű —, Manhattan Project: atomfegyverek építésének terve, SETI program: földön kívüli értelmes lények felkutatását célzó akció, a nők választójoga(!?), az ENSZ hivatalos helyeinek listája. Lekérhetjük a világ öt legnagyobb városát, a civilizációban legelőrehaladottabb nemzetek rangsorát, a világtérképet, az űrhajókat (ha vannak), és egyéb demográfiai tablókat (népesség, nemzeti jövedelem, az ország területe, az írni-olvasni tudók aránya, betegségek, környezetszennyezés, várható életkor, családméret, katonai szolgálat, a gazdaság termelékenysége stb.)

A véletlenül fellépő természeti és egyéb katasztrófák: földrengés, éhínség, tűzvész, árvíz, kalóztámadás, pestis, vulkánkitörés bármikor jelentős lélekszámveszteséggel fenyegetik városainkat. A programban rendelkezésünkre áll egy nagyszerű kislexikon, a Civilopédia, itt minden egyes fogalom, technikai és társadalmi vívmány pontos angol nyelvű leírását megtaláljuk, külön kiemelve, hogy ez milyen más mutatókat befolyásol, illetve ennek megjelenéséhez melyek a feltétlen kritériumok.

A városok egy természetes fejlődésen mennek keresztül, de ez a folyamat kincstári segédlettel felgyorsítható. Előbb építtethetünk magtárakat, laktanyákat, templomot, piacot, könyvtárat, vízvezetéket, bankokat, bírósági épületet, stadiont, hajókat, a város köré védőfalakat, gyárat, palotát, erőművet, újrafeldolgozó üzemet (recycling center), egyetemet, űrvédelmi eszközöket (SDI — Strategic Defense Initiative), a világ csodáinak bármelyikét, előbb toborozhatunk seregeket, telepeseket, mint ahogy azok maguktól kialakulnának. Persze a város el is adhat vagy bérbe adhat épületeket, ha plusz bevételre van szüksége.

A kőművesség és a magfúzió

Az érdekesség kedvéért még néhány fontosabb lépcsőt is felsorolunk, mely az emberiség életében mérföldkőnek tekinthető: a kőművesség, az írás, az asztronómia, a filozófia, a puskapor, a vegyészet, a robbanóanyagok, az elektronika, a komputerek, a génsebészet, a műanyagok, a szupravezető anyagok, a nukleáris és magfúziós energia felfedezése.

Ugye, magas ívű szellemi repülésre biztatjuk azokat, akiket megfogott eme nagyszerű játék ötletessége?

Ugye, már rájöttek a címben feltett kérdés értelmére? Mert aki még nem civilizált (a Civilizationnal), az nagyot mulasztott. Próbálja ki Ön is! (Még a földtörvény meghozatala előtt!)

A program (Sid Meier's Civilization, Microprose) ára az Alaplap Postában: 6400 Ft + ÁFA.

A Maxis Software sokat adott, mégis adós

Tavaly már egy ízben részletesen ismertettük a SimCity nevű városszimulátort (lásd 91/5., 37.old.). Ezúttal a program Microsoft Windowsos változatára szeretnénk felhívni a figyelmet. Előrebocsátjuk, hogy a játékszabályok semmiben sem változtak, minden maradt a régiben, azonban az összes manipuláció legalább ötször kényelmesebb lett.

Minden beavatkozási lehetőség csinos kis ikont kapott, így egérrel azonnal elérhető. Még a PC szerény képességű, egycsatornás hangszóróján is monumentálisan szóló zenei alapmotívumot kapott a program, valamint a bejelentkező képernyőn Maxine, a tehén mosolyog kedélyesen az ábrázatunkba. Az új installációs eljárás képes a SimCity Graphics 1 és 2 „városdíszletező" programok beolvasására is, így a hagyományos építőelemek helyett azonos szabályok mellett építkezhetünk az ókori Ázsiában, a középkori Európában vagy a vadnyugaton, illetve a 2000-es évek USA-jában, 2050 Európájában vagy éppen egy kolónián a Holdon. Megjegyezzük, hogy ez a program már nem tartalmaz védelmet, így nem kell megvárni a feltört változatát sem.

Mindenképpen érdemes áttérni rá, azaz „polgármester kerestetik"!

(Erről eszünkbe juthat — tudva, hogy a Maxis Software elkészítette a nagy sikerű SimEarth bolygószimulátorának a Windowsos változatát is —, hogy más is kerestetik! A hangyaboly-szimulátor — SimAnt — mellett kíváncsian várjuk a RoboSport for Windows nevű új dilit, amely transzformerek és egyéb, nem emberszabású géplények végtelen változatosságú harcaival borzolja — vagy nyugtatja? — idegeinket.)

A K C I Ó !

Megrendeléstől a kulcsrakész átadásig: *72 óra!*



IRODA KULTÚRA STÚDIÓ



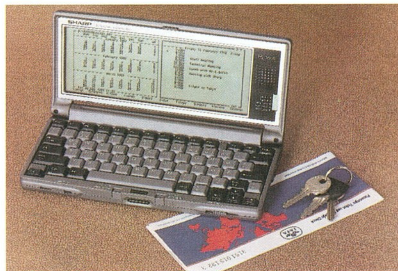
1067 Bp. VI., Podmaniczky u. 27.
Tel.: 131-8168, Tel./Fax: 132-0188

Bemutatóterem:
1054 Bp. V., Kálmán I. u. 14. Tel./Fax: 153-4898

Vidéki iroda:
Pécs, Szalay A. u. 12/A Tel/Fax: (72) 21-181

INFORMÁCIÓKÉRÉS: 22 ▼

EZT ZSEBRE TEHETI...



A PC 3100-as palmtop = IBM PC + menedzserkalkulátor

- 640x200 LCD display
- 2 MB RAM
- 1 MB ROM
- RS 232C és Centronics interface
- 77 gombos billentyűzet
- DOS 3.3 } ROM
- Laplink szoftver } ROM
- CPU: 80C88A/10 MHz

Opciók:

- 2 db PCMCIA 1.0 IC memóriakártya
- 1 db 3.5"/1.44 MB FDD
- Áramellátás: 3 db ceruzaelem (LR 6)

A készülék súlya mindössze 480 gramm!

Fogyasztói ára: 90.000,– Ft + ÁFA, REKLÁMÁRON!



Kapható:
EURO-PROFIL
1147 Budapest XIV., Fűrész u. 65/b
Tel.: 163-5210, 163-6095

INFORMÁCIÓKÉRÉS: 24 ▼



...gyors emelkedés

PERON · REKLÁM Kft.

Telefon: 149-4819

INFORMÁCIÓKÉRÉS: 23 ▼



ICS

FESTÉKSZALAG

és

CÍMKE

minden Magyarországon működő

VONALKÓDNYOMTATÓHOZ.

Rendkívüli ajánlat: minden második

I N G Y E N

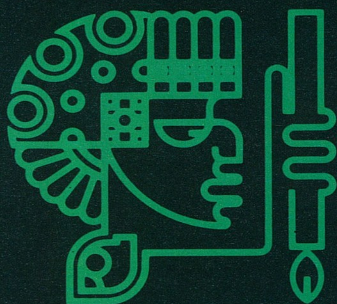
van!

Mag ICS

Informatikai Rendszerfejlesztő és Marketing Kft.

H-9400 Sopron, Bástya u. 75. Tel./Fax: (36) 99/14-250

INFORMÁCIÓKÉRÉS: 25 ▼



incas®

Bármilyen festékkazettát keres, nálunk megtalálja!

CÉDRUS KAROLINA ÁRUHÁZ

Budapest XI., Karolina út 17. • Tel.: 166-2111 • Fax: 185-2221

INFORMÁCIÓKÉRÉS: 26 ▲

Prix Ars Electronica 1992, Linz

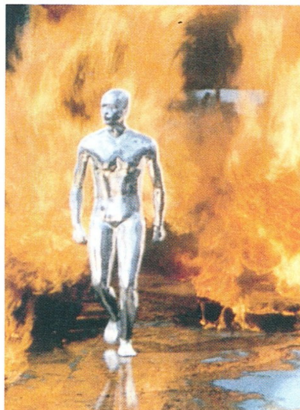
Átlépés egy másik dimenzióba

A rendezvényről hazatérve azzal összegeztük élményeinket, hogy ez egy óriási kiállítás volt. No nem a méreteit, hanem a bemutatott alkotásokat tekintve. A művészi teljesítményhez pedig kitűnő hátteret nyújtott az, hogy az egyik fő szponzor a Silicon Graphics, amely manapság a legjobban használható RISC-processzoros grafikai munkahelyeket gyártja.

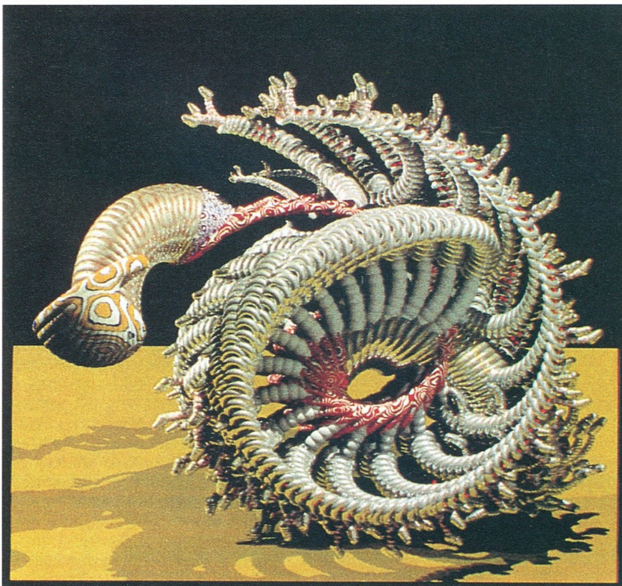
Ezt az egyidejűleg több helyszínen futó linzi kiállítást és rendezvénysorozatot 1987 óta évenként megrendezik, jelentős összegű díjakkal jutalmazva a legjobbakat. Ebben az évben például 1,25 millió schillinget osztottak ki, megtoldva az erkölcsi siker értékével. Csak egyetlen példa: 1987-ben a zenei díjat Peter Gabriel, a Genesis egykori alapítója és énekese kapta. Hazai és magyar származású külföldi alkotók is szép sikereket értek már el az elmúlt években.

A verseny jelenleg az alábbi 4 kategóriában zajlik:

— Számítógépes grafika
— Számítógépes animáció
— Interaktív művészetek
— Számítógépes zene



Dennis Muren—Mark Dippé—Steve Williams: Terminátor II

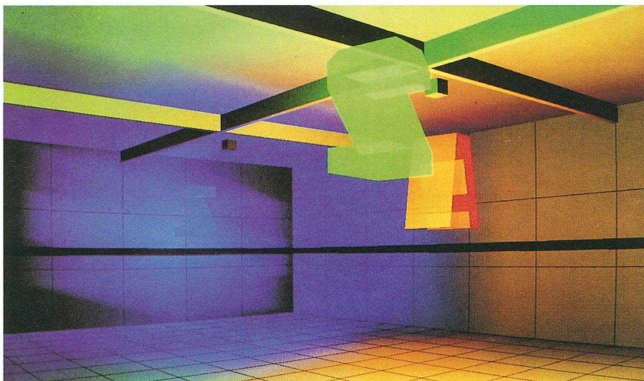


William Latham: Fénysugárkövetéses változatok a végtelenség repülőgépén

Virtuális (látszólagos) valóság

Számomra az idei rendezvény egyik fő érdekessége a VR, az úgynevezett „virtual reality" volt. Képzeljük el, hogy kap az ember egy furcsa szemüveget, amelyben színes LCD képernyő van. Ezen láthatók a számítógép által közvetített képek. Ehhez eredetileg még egy fejhallgatót is kapunk (bár a kiállításon ezt már lespórolták), amin a virtuális világ hangjait hallhatjuk. Végezetül a jobb kezünkre ráhúznak egy kesztyűt, amely érzékeli kezünk és ujjaink mozgását, s amellyel megadhatjuk, hogy merre akarunk haladni.

Ennyi technikai előfeltétellel tehát „beöltözik" az ember a „virtuális valóságba". Három dimenzióban élhetünk át egy olyan világot, amely elszakad hétköznapi valóságunk dimenzióitól. Ahogy forgatom a fejem, leguggolok, felegyenesedek, úgy változik a képzeletbeli táj nézőpontja, mintha csak valódi lenne. Ahogy mozgatom a kesztyűs kezem, arra haladok, amerre az ujjam mutat, gyorsíthatok, lassíthatok, megállhatok, visszafordulhatok. Benézhetek például egy kúp alá, amely a föld felett öt centire lebeg, láthatok mindenféle furcsa spirális képződményt, óriási emberi fejet, természetesen fejjel lefelé... meg mindent, amit az alkotói



Jeffrey Show: Virtuális Múzeum — 5. terem

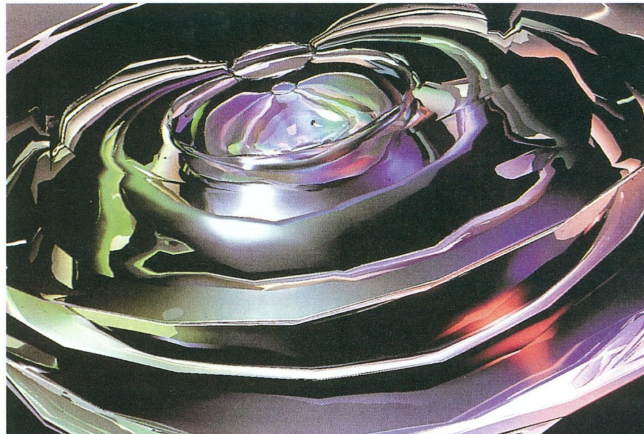
fantázia kitalált. A valós világgal szembeni különbség eleinte ijesztő, de meg lehet szokni... utána pedig az ember alig talál vissza.

Lássuk azért a dolog technikai oldalát is! A két szem számára természetesen külön-külön képet kell létrehozni, realtime módon. Az óriási számítási és képfeldolgozási sebességet két külön Iris Crimson (Silicon Graphics) biztosítja. Egyik az egyik szemre és fülre dolgozik, a másik pedig a másikra. A kettőt normál számítógéphálózaton keresztül egy Macintosh köti össze. Van továbbá az érzékelőkről bejövő jeleket feldolgozó, és a számítógép számára megfelelő formátumúra hozó három fekete doboz, s azok rettenetes kábeldzsungelen keresztül vannak összekötve a Mac-kel. Végezetül egy óriási projektor is tartozik a mutatványhoz, hogy a kívülállók is élvezhessék ezt a háromdimenziós utópiát.

Virtuális múzeum

Továbbmenve láthattunk az előzőhöz hasonló elvű „múzeumot". Egy forgó korong áll egy üres terem közepén. A korong egyik szélén fotel, szemben vele pedig egy kivetítő, alatta a sokat sejtető Silicon Graphics 4D/310 VGX szupergéppel.

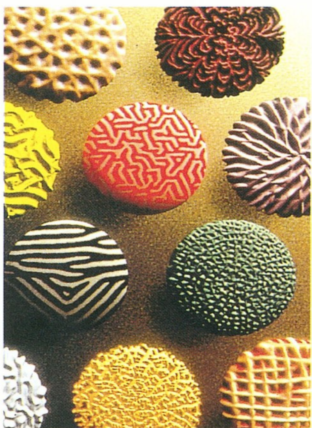
Ha az ember beül a fotelba, és elkezd oldalra fordulni, az egész forgókorong vele fordul, a képernyőn megjelenő kép pedig, (amely egyébként valódi múzeumi tárgy mása) úgy változik, mintha tényleg elfordultunk volna egy valódi teremben. Ha a fotelban előre dőlünk, ezzel a múzeumban előre megyünk, ha hátra dőlünk, akkor pedig hátra. Ha elég közel vagyunk egy ajtóhoz, előre dőlve átsétálhatunk a másik terembe. Az igazi látványosság mégis az, amikor meglátjuk az egyes termek „életét". Az egyikben óriási világító számok ugrándoznak a padló és a plafon között, a másikban mindenféle kínai írásjelek jönnek ki a falból az egyik oldalon, s tűnnek el a másikon. Az egész olyan érzetet kelt, mint annak idején a Beatles együttes „Sárga tengeralattjáró" című rajzfilmje.



Stewart McSherry: Chevlot

„Szemforgatás"

Szintén nagyon érdekes és magyar vonatkozású virtuális világ volt Agnes Hegedüs „Handsight" című alkotása.



Andrew Witkin—Michael Kass: Reakció-diffúziós gombfelületek. (Részlet)

Ez az installáció kevesebb helyet foglalt el, mint az előzőek, leszámítva a szoba sarkában meghúzódó Silicon Graphics Power Series 4D/310VGX gépet. Az egész tulajdonképpen egy biliárdgolyó nagyságú „szem", amit kézbe fogva egy kb. 40 cm átmérőjű, felül nyitott üveggömbben lehet vele „nézelődni", azaz a „szemmel" belenyúlni a gömbbe, és ott „szemet forgatni". Az üveggömb a valóságban természetesen üres, de a kivetített képen úgy látszik, mintha egy halszem-optikával nézve sokféle tárgyat — szerszámot, rugót, harapófogót, kalapácsot — vennénk szemügyre. Úgy változik a kép, ahogy a kezünkkel mozgatjuk a gömbben a „szemet".

Egészen furcsa alkalmazást mutatott be egy német szerzőpáros. A falon egy tévéképernyő volt, képkerettel bekere-

tezve, a képernyőn pedig egy németalföldi reneszánsz mester képe, és „első ránézésre" az egész úgy nézett ki, mintha igazi festményt látnánk. A trükk akkor működött, ha leültünk a képpel szemben, és elkezdtük nézegetni a „festményt". A kép ugyanis furcsa módon mindig ott maszatolódott el, ahová éppen néztünk. De honnan tudhatja egy számítógép, hogy a kép melyik részét vesszük szemügyre? A megoldás a kép alatt elhelyezett, félig áteresztő tükör, amely mögül egy infra kamera a székben ülve a képet figyelő ember szemére volt ráállítva. Egy kis kalibrálás után a kameráról lejövő kép, amely az ember szemmozgását mutatja, „könnyen" konvertálható koordinátákká, annak alapján pedig már igazán semmiség valamilyen módon megváltoztatni a képet azon a részen, ahová a szemlélő néz.

Filmeffektusok

Szóljunk egy olyan műről is, amelyet nem állítottak ki, azonban mint referenciáról a kiállítás összes anyaga említést tesz róla. Ez pedig Dennis Mure, Mark Dippé és Steve Williams alkotása James Cameron filmjében, a Terminátor II-ben. (Amikor ezt a cikket írom, az MTV-n éppen Arnold Schwarzenegger veszi át a legjobb férfiszereplőnek járó díjat a fenti filmben nyújtott alakításáért).

Aki látta a filmet, biztos, hogy nem felejti el, amikor a T1000-es átfolyik a börtönrácson, vagy amikor „felkel" a kockaköves folyosóról és felveszi az őr alakját. Ezeket az effektusokat természetesen szintén a Silicon Graphics gépeivel hozták létre. Pontosabban szólva harmincnál is több Iris 4D grafikus munkaállomás dolgozott a film képeinek létrehozásán, s azokon futott a rengeteg szoftver, többek között az Alias 2.

A klasszikus műfaj

A Prix Ars Electronica pályázat és kiállítás egyik hagyományos ágazata a számítógépes grafika. Talán ezen mérhető le legjobban a számítástechnika művészeti eszközként való felhasználásának fejlődése. Az első időszakban ugyanis sem az interaktív művészet, sem az animáció, sem a zene nem volt jellemző a számítástechnikai alkalmazásokra, legalábbis nem a nagyközönség számára elérhetően. A grafika viszont kezdettől fogva élénken foglalkoztatta a számítástechnikusok és a művészek fantáziáját. Viszonylag könnyen hozzáférhető gépekkel is tetszetős grafikai műveket lehetett alkotni, legfeljebb a számítási idő volt hoszszabb, a felbontás és a színek száma pedig kevesebb.

Az Ars Electronicán Andrew Witkin és Michael Kass nagyon szép, valósághű grafikákat mutatott be, természetesen Silicon Graphics gépen, saját fejlesztésű szoftverrel készítve.

Stewart McSherry kiállítási anyaga viszont azt mutatja, hogy az alkotó a tükröződő és áttetsző felületek ábrázolásában éli ki magát. Ehhez a Silicon Graphics 4D/210 gépen futó Alias 3.0 raytracer (fénysugárkövető) szoftverre is szüksége volt a felületek fénytörési és fényvisszaverési tulajdonságainak visszaadásához.

De nem minden grafikához szükséges feltétlenül a Silicon Graphics valamelyik számítógépe. Itt van például Bullás József és Krajcsik Zsolt grafikája, amely bonyolult textúrájú felületeket ábrázol, és Compaq 486-os gépen készült. Persze egy kis grafikai bővítés kellett hozzá, hogy a felbontást és az árnyalatok számát megfelelően lehessen bővíteni. Ezen a képen is látszik, hogy elkészítése nem két percig tartott.

Egy másik „PC-alapú" grafika Yoshiyuki Abe absztrakt, tükröződő felületeket ábrázoló alkotása, amely talán az egyik legtetszetősebb, és szintén 486-os gépen készült, fénysugárkövetéssel.

William Latham viszont inkább az IBM 3090-re esküszik, de valószínűleg azért, mert „ez volt kéznél". Ilyen gépet senki nem tart otthon a lakásban, kivéve talán az őrült milliomosokat. (Nem mintha a Silicon Graphics nagyobb gépei elérhetők volnának a magánember számára.) Ez az IBM 6090-es képernyőn bemutatott grafika egészen élethű százlábú szörnyet ábrázol, de az sem baj, ha valaki másra asszociál.

Mitől mozog?

Az óriási „mozgó" látványosságokat itt nem tudjuk illusztrálni, pedig azok között is volt néhány figyelemreméltó alkotás. Például Karl Sims animációs produkciója, amelyhez (szerényen) egy Connection Machine System CM-2-es gépet használt. Erről a gépről tudni kell, hogy minimum 65536 (!) processzort tartalmaz, s azokat sejtprocesszorként lehet programozni, Lisp programozási nyelven. Ezek után elképzelhető a gép számítási sebessége. Ami viszont nem képzelhető el, az az ára. Nem is „futkosnak" ezerszámra a világon ezek a gépek, és Karl Sims sem otthon játszadozott, hanem munkája során a Massachusetts Institute of Technologyn futott össze vele. Az említett animáció készítésekor egy kiinduló képre az Artificial Evolution programmal ráengedett számítási algoritmus a képből valamilyen „mutánst" hozott létre. Kissé módosított algoritmussal azután mindig újabb mutáns keletkezett. Most képzeljük el az egészet úgy, hogy minden egyes képponthoz egy processzort rendelünk, hadd foglalkozzon külön vele. Az újabb mutációk létrehozására tehát nem kell sokat várnunk. Amikor a művésznek valamelyik megtetszik, azt veszi kiinduló képnek, mintegy beavatkozva az „evolúció" menetébe. A végeredmény egy 2 perc 15 másodperces, valamint egy 1 perc 35 másodperces animáció. Egyszerű, nem?

Illúziókeltés és -rombolás

Mint minden hasonló rendezvényen, az az Ars Electronicán is voltak szegényes, anakronisztikus, és oda nem illő alkotások, de nem ez volt a jellemző, s azokkal nem is foglalkozom. Szóljunk inkább egy-két érdekes installációról.

Tetszett például Christian Möller „Space Balance" című alkotása, egy sötét teremben lévő libikóka, amely — ha az ember középen ráállt — ide-oda billegett. Az egésznek a lényege viszont az, hogy a terem két szemben lévő falára vetített kép a libikóka meghosszabbítását mutatta, mégpedig úgy, mintha az egész terem velünk együtt mozogna. A teremben még kisebb-nagyobb golyók is voltak, amelyek szintén össze-vissza gurultak, sőt hangot is adtak, amikor visszapattantak a falról.

Igen érdekes volt az a kiállítás is, ahol régi idők analóg szintetizátorait mutatták be, például a Moog és ARP gépeket. Ezeket ki is lehetett próbálni, s ettől természetesen hatalmas hangzavar volt az egész múzeumban.

Végül néhány szót két egymást követő „performance"-ról. Az egyiknek „IMmediaCY" volt a címe. Őszintén szólva szörnyű volt. Laurie Andersont akarta utánozni, kevés sikerrel. Már a címére sem emlékszem viszont a másiknak, ahol MIDI-vezérelt hegedűből csiholtak elő rettenetes hangokat, és ezeket össze-vissza képekkel szinkronozták. Akinek tetszett, lelke (szeme? füle?) rajta!

Összegezés helyett csak annyit tudok mondani, hogy jövőre is kimegyek az Ars Electronicára. Ha valahol, akkor itt képet kaphatok arról, hogy hol tart a számítógépes-elektronikus művészet.

Sík Zoltán

Az MPIR páciens-nyilvántartó információs rendszer

Organikus, permanens folyamat

Az alább vázolt elvek alapján a szerzőknek sikerült körülbelül másfél évi folyamatos fejlesztéssel egy hasznos, stabilan működő, számos vonatkozásban előremutató, „egészséges" rendszert kidolgozniuk. A cikket különösen ajánljuk a medicina területén dolgozó, rendelőkben, kórházakban foglalkoztatott olvasóinknak, de a magánpraxist folytató doktoroknak is — mellettük azonban mindazoknak, akik szakértő (jellegű) rendszerek iránt érdeklődnek.

Az olyan bonyolultságú fejlesztések, mint amilyen például egy, az orvosi munkát támogatni kívánó számítógépes rendszer, nem végezhetők a hagyományos, skolisztikus módszerekkel — azaz egy felmérő-analizáló munkafázist követő, „végleges, teljes, kimerevített" programrendszer kidolgozásával. Ugyanis:

- dinamikusan változnak a feltételek, bővülnek az alkalmazható eszközök (hardver, szoftver);
- gyakorta módosulnak az ügyviteli, elszámolási eljárások, előírások;
- az adott fejlesztés realizálása nyomán törvényszerűen alakul az alkalmazásról, a rendszerről kialakított kép, pontosabb lesz a feladat- és célrendszer;
- a nehezen, illetve csak részben strukturálható, de meghatározó, döntő fontosságú szakmai, orvosi ismeretek, munkamódszerek (diagnosztizálás, terápiabehatárolás, kiértékelés stb.) számítógépes támogatási lehetőségei is permanensen változnak, finomodnak a fejlesztésekkel. (A fejlesztési zsákutcákból sokszor többet lehet tanulni, profitálni, mint a sikerekből.)

Mi olyan fejlesztési eszközöket (jórészt saját CASE-jellegű bővítésekkel) használunk, amelyekkel a gyakorlatban a változásokkal, a felvetett észrevételekkel, igényekkel szinkronban végezhető a rendszerfejlesztés anélkül, hogy közben a felhasználónál valamiféle hátrány (adatvesztés, időbeli kiesés) jelentkezne. Nem „végleges", előzetesen „agyonelemzett" megoldásra törekszünk, hanem egy működő, és a gyakorlatban gyorsan verifikálható és módosítható megoldás kialakítására. Eközben szorgalmazzuk a minél szélesebb körű alkalmazásbavételt, és a felhasználók észrevételeinek azonnali visszacsatolását a fejlesztésbe.

Amikor a rendszert valamilyen fórumon bemutatjuk, a hazai szoftverfejlesztői szakma képviselői általában megkérdezik, hogy milyen fejlesztőeszközzel dolgozunk. A választ hallva pedig, hogy az MPIR moduláris felépítésű rendszert a Clipper relációs adatbáziskezelőre alapozva dolgoztuk ki, mindig felvetődik:

- Melyik a legkorszerűbb, a leghatékonyabb eszköz?
- Miért éppen ezzel fejlesztünk, miért nem egy újabbal dolgozunk?

A mi „lovunk"

Az MPIR rendszer fejlesztése hét évvel ezelőtt kezdődött el a dBase/Clipperre alapozva. Akkoriban PC-n ez volt a „legkorszerűbb" eszköz. Az egyre összetettebb feladatok megoldásán keresztül alakult ki az a saját fejlesztői programcsomag, valamint jött létre az az ismeretbázis, amellyel a mai feltételek mellett is hatékonyan tudunk dolgozni. Természetesen jól tudjuk, hogy az újabb, fejlesztést támogató eszközökkel kezdő szinten is gyorsan és megbízhatóan lehet rugalmas nyilvántartási rendszereket létrehozni, és mi magunk is intenzíven foglalkozunk — már munkakörünknél fogva is — az újabb és legújabb eszközök alkalmazásbavételével. Az az „üzemszerű" állapot, hogy a fejlődéssel meg kell tanulni együtt élni, együtt dolgozni. A cselekvést, a konkrét fejlesztői munkát azonban nem pótolja semmi, és a bevált szerszámot a legtöbb esetben nem érdemes cserélgetni.

Szakosodás/szakértés

Anélkül, hogy itt részletesebben belemennénk a szakértő rendszerek igen sok nyitott kérdést tartalmazó problematikájába, néhány átfogó megállapítást kívánunk tenni.

A modern értelemben vett számítástechnika alkalmazásának kezdetein már kidolgoztak olyan programokat, amelyek a fizikai, műszaki rendszermodellezés területén az ún. optimális megoldások, rendszerállapotok és beavatkozások behatárolását sikeresen oldották meg. Azonban ezeket az analitikus, numerikus, klasszikus értelemben vett matematikai eljárásokat már az összetetteb fizikai, műszaki rendszereknél sem lehetett alkalmazni, hanem sajátos heurisztikus keretbe ágyazott „kevert", egyedi problémafüggő programok kerültek kidolgozásra (szimulációs, sajátos problémafüggő/elemző, kereső eljárások).

Az összetett, rosszul strukturált, dinamikus gazdasági, társadalmi, biológiai, orvosi stb. rendszereknél, feladatoknál pedig gyorsan túllépte az idő a misztifikált „automatikus, optimális" számítógépes fejlesztési törekvéseket. Helyettük a számítógépes „támogatás, segítés", valamint a komplex, bonyolult feladatoknál a „szakértő rendszerek" fogalma és ezekkel kapcsolatos irányok léptek.

Meglátásunk szerint a realizált újabb szakértői rendszerek mindig valamilyen relációs adatbázis-kezelőre épülnek — figyelembe véve a jelentős és bonyolult adatfeldolgozási igényeket —, az alapvető feladatot azonban az adott, többnyire „élő", nehezen strukturálható („fuzzy") adat-, ill. ismeretmodell, valamint a feldolgozás módja jelenti. A problémák modellezése döntően heurisztikus, alkalmazva a komplex döntések területén feltárt eljárásokat, azonban ezeket sem szolgaian. Elsősorban mint megközelítési mód dominál ez: kreatívan, nagyfokú adaptivi-

tással, kihasználva az adott feladat sajátosságaiból származó lehetőségeket, és figyelemmel a rendszert használókra.

Az MPIR rendszer döntéstámogató moduljai is a leírtak szellemében készültek.

A rendszer három fő alkalmazói modulcsoportból áll, amelyek fejlesztését és forgalmazását is különböző stratégia szerint alakítottuk.

Az alapmodulrendszer ingyenes, szabadon másolható. Ennek ellenére adatnyilvántartási szempontból korlátozások nélkül használható. Tartalmazza az alapadat-nyilvántartást és a páciensekhez kapcsolódó kezelések nyilvántartó és lekérdező eljárásait, valamint az adat- és rendszerbiztonsági modulokat. A tapasztalatok szerint ez mint alaprendszer önmagában is jól használható.

A felhasználó a rendszer ún. regisztrálásakor számos, saját adataival specifikált modulhoz jut, s ezenkívül külön megveheti a számára szükséges egyéb bővítéseket.

Az MPIR a fentieken kívül tartalmaz komplex, expert jellegű alkalmazói modulokat is: döntéstámogató, szakértői (géppel támogatott diagnosztizálás, terápiaajánlás stb.) eljárásokat is, amelyek jelenleg gyakorlati tesztelés alatt vannak (magánpraxisban, oktatásban, kutatásnál). E cikkben azonban csak a főbb részek általános leírására szorítkozunk.

Szeretnénk itt megjegyezni, hogy mi csak a lehetőségét teremtettük meg annak, hogy a program PC-s környezetben, a gyakorlati munkában is képes legyen érdemi segítséget nyújtani. Ami a továbbiakban is alapvető, és a gyakorlatban hatalmas és folyamatos munkát jelent, az az ún. ismeret- és tudásbázis bevitele a rendszerbe. Ez még kiegészítésre, pontosításra szorul, és szívesen vennénk olyan orvosok, orvoscsoportok jelentkezését, akik részt vennének a speciális modulok kidolgozásában, pontosításában, tesztelésében.

Kartonok modul

A betegek alapadatainak rögzítésére szolgál, gyakorlatilag a kartotékrendszert váltja ki. Sokféle szempont szerint támogatja a gyors visszakeresést. Minden kartonhoz kapcsolható kb. száz A4-es oldalnyi szöveges információ. A több szempontú logikai keresést ezekben a leírások is erősen támogatják. Továbbá lehetőség van arra is, hogy a páciensek tetszőleges számú és megnevezésű logikai csoportokba sorolhatók legyenek. Az így besorolt betegek közül bármelyik listánál válogathatunk: „csak azok szerepelnek rajtuk, akikre igaz hogy...". Ugyanezzel a válogatási lehetőséggel tehetjük meg azt is, hogy a kartonok közül csak azokat másolja ki a rendszer lemezre, amelyeket meghatároztunk. Ez az adatmentési lehetőség tulajdonképpen egy postázó modul. A beteg összes adatát kérésre kimásolja lemezre, és így az hordozható. Átadható másik egészségügyi intézménynek, vagy akár hatóságnak.

Kezelések modul

Ebbe a táblába, mint egy pénztárgépbe, rögzíthetők azok az adatok, amelyek alapján a hatóságok felé generálhatók a szükséges statisztikák és listák, a legkülönbözőbb bontásban és sorrendben.

Betegségek modul

A BNO-kódok listája mellett még számos hasznos adatot tartalmaz, továbbá a betegségek rövid algoritmikus leírását is.

Vizsgálatok modul

A betegeken végzett vizsgálatokat és azok eredményeit tárolja itt a rendszer. Elővalogatott csoportok teszik kényelmessé és gyorssá az orvos munkáját.

Kinyomtathatók a labornak szóló levelek, de — ha hálózatban működik a rendszer — nem az orvosnak kell az eredményeket beírnia, hanem azt a labor munkatársai végzik el, és a következő másodpercben már le is kérdezhetők az osztályon. A labor dolgozói nem tudják név szerint, kinek a vizsgálatairól van szó. Ezzel támogatni kívánjuk az orvosi titoktartás érvényesíthetőségét, és a személyi jogok védelmét. Normálérték-szótár is tartozik ehhez a táblához, mely azonnali segítséget ad az adatok kiértékeléséhez és viszonyításához. Gombnyomásra grafikusan, idősorba rendezetten jelennek meg a beteg vizsgálati adatai a képernyőn, így az orvos szemléletesen láthatja visszamenőleg is az egyes értékek alakulását.

Diagnosztizálást támogató modul

Ez a modul már több helyen váltott ki érdeklődéssel párosuló vitát. Nem véletlenül, hiszen szerte a világon folynak fejlesztések és kutatások ezen a területen, és őszintén szólva tartottunk egy kicsit az orvosi szakma kedvezőtlen hozzáállásától. Különösen a hasonló témát kutató szakemberek óvtak bennünket a gyakorlatban dolgozó kollégáiktól. De furcsa (vagy talán nem is furcsa) módon éppen a gyakorló orvosok részéről tapasztaltunk olyan aktív érdeklődést és támogatást, amely nem csak inspiráló és erőt adó, hanem meghatározó módon hasznosítható volt. Erre azonban még sokáig szükségünk is lesz — itt magunk nem boldogulunk.

Tréning modul

A program egy képzeletbeli beteget generál az alábbi módon:

1. Véletlengenerátor segítségével egy konkrét betegségre utaló tünetsorozatot állít elő, normál adatokkal vegyesen.
2. A felhasználó eldöntheti, vagy a géptől kérheti azt, hogy az adott értékek magasak, alacsonyak vagy normálisak-e.
3. Meg kell határozni a valószínűsíthető betegségeket.
4. A programrendszer a benne lévő tudásbázis alapján magyarázatokkal egyetemben felsorolja a szóba jöhető betegségeket, csökkenő valószínűséggel, illetve bizonytalansági tényezőkkel.
5. A felhasználó meghatározhatja az általa alkalmazandó terápiát.
6. A programrendszer is megadja a javasolt gyógymódokat.

Ez a modul az oktatásban, a kutatásban, a rendszer tesztelésében, ellenőrzésében és a benne lévő ismeret-/tudásbázis finomítására alkalmazható.

Gyógymódok modul

Itt tárolhatók az alkalmazandó gyógymódok. Értjük ezen az európai, a keleti, vagy akár a természetgyógyászatban alkalmazottakat is, korlátozások nélkül. A gyógymódajánló modul képes arra, hogy ezek közül elővalogassa az orvos számára azon gyógymódokat, amelyek az adott betegség(ek)nél szóba kerülhetnek, de továbbra is az orvosnak kell ezeket bevizsgálnia, értelmeznie, majd dönteni. Mindenesetre a rendszer felhívja a figyelmet arra, hogy

- mely gyógyszereket szed most a beteg, milyen gyógykezelés(ek) alatt áll;
- mely gyógyszerekre, gyógyszercsoportokra érzékeny, melyek ellenjavalltak;
- melyek állnak egymással kölcsönhatásban;
- van-e köztük olyan, amelynek alkalmazásbavételéhez további vizsgálatok szükségesek (természetesen, ha elkészülnek a kérdéses vizsgálatok, akkor ellenőrzi azokat);

• tájékoztatást ad az adott gyógymódok szokásos adagjairól;
• elolvashatóak a rövid információk (mint például a „zöld könyvben").

Még számos információ található a gyógymódokról, így például az ár, a felírásával és alkalmazásbavételével kapcsolatos előírások és szabályok, a hatáskódja, kiszerelései stb. Ezen információkat természetesen nem mi, a fejlesztők tápláljuk a rendszerbe, hanem megfelelő információkkal rendelkező cégek, intézmények, például a HungaroPharma. Meggyőződésünk, hogy az adatokat azoknak kell karbantartani, akik azokkal elsődlegesen kapcsolatban állnak.

A rendszer képes a vények szabályos kiállítására és kinyomtatására is.

Határidőnapló

A határidőnapló is hasznos szolgálatot tehet a programcsomagot használó számára. Abban a pillanatban, amikor egy beteget berendel az orvos, egy elfoglaltsági tábla jelenik meg a képernyőn. Itt láthatóak a még szabad időpontok, és nem kell fellapozni a naptárat. A program ezt automatikusan vezeti, de felírható ide az ügyelet, a szabadság, a továbbképzések, és tulajdonképpen mindenféle időbeli elfoglaltság.

Kórházi, ill. többfelhasználós környezetben lehetőség van arra, hogy közös, szabad időpontokat keressünk, pl. egy megbeszélés időpontjának kijelöléséhez.

Eszközök

Ez egy egyszerűsített készletnyilvántartó modul, melyben nyomon követhetőek az egyes tételek mozgásai; például hogy ki, mikor, kinek és mennyiért adott vagy vett át valamit, hogy mennyi a pillanatnyi készlet stb.

A futtatási környezet

Minimál konfiguráció — egygépes felhasználásra:

1db IBM XT/AT-kompatibilis számítógép:
• központi tár: 640 kbájt,
• 1 db floppyegység,
• 1 db merevlemezes egység,
• monitor: monokróm vagy színes,
• néhány floppy az adatok mentéséhez,
• DOS 3.3 operációs rendszer.

Az MPIR rendszer működik többgépes, hálózatos környezetben is, ezáltal alkalmas körzeti rendelők, korházi osztályok munkájának komplex támogatására is.

A rendszerben a felhasználó a megszokott szakmai, alkalmazói kifejezéseket használhatja.

A menüvezérléssel és a help szolgáltatással önmagát tanítja, magyarázza a rendszer. (A helpet a felhasználó átírhatja, bővítheti.)

A többszintű jogosultság (ún. password-hozzáférés), az egyszerűen vezérelhető rendszerkarbantartási funkciók az adat- és rendszerbiztonságot szolgálják. A korszerű adatbázis-kezelés és fejlesztő eljárások biztosítják a rugalmas bővíthetőséget, az egyszerű módosítást, a továbbfejleszthetőséget.

Végezetül (és némiképp célzatosan)

A rendszerben található egy message-box is. A rendszerben dolgozó felhasználók üzenhetnek egymásnak még akkor is, ha a címzett nincs éppen a helyén. Majd megkapja, amikor legközelebb „belép". Még egyfelhasználós környezetben sem haszontalan, hiszen a délelőtti váltás üzenhet a délutáninak. A mi munkahelyünkön azonban, ahol fut már néhány hálózatos alkalmazásunk — az „SLIST" többnyire 35-40 működő szervert mutat —, a felhasználók előszeretettel használják mindennapi munkájukban a message-box szolgáltatásait. Ezen keresztül kapjuk mi is az észrevételeiket. Üzeneteket azonban másképpen és máshogyan is várunk...

Kiss János — Keresztúri János

Forrasztópáka helyett alkatrészcsere

A hardver(es) jövője

El vagyok keseredve.
Mint hardverrel is foglakozó szakembernek, megvan rá az okom, amint azt az alábbiakban elmondom. Lehet, hogy nem mindenki ért velem egyet, de egy kicsit talán ők is együttéreznek velem.

Aki megteheti, nézzen bele egy „régi" XT vagy PC, majd egy 486-os gép belsejébe. Az alaplapot szemügyre véve látható, hogy az XT-ben szereplő áramkörök mindegyike az elektronikai tervezésben általánosan használt, jól ismert ún. katalógus-áramkör (mivel katalógusokban szerepel). A szerelés is a megszokott, nyomtatott áramkörös megoldású. Ez azt jelenti, hogy az integrált áramkörök, kondenzátorok és ellenállások kivezetéseit a nyomtatott áramköri lapon fúrt lyukakon átdugva és beforrasztva rögzítik.

Egy 286/386/486-os alaplap már egy más világ: igen finom rajzolatú, többrétegű nyomtatott áramköri lemezen kialakított, felületszereléses technológiával készülnek ezek, és speciális felhasználóorientált áramköröket tartalmaznak. A felületszereléses technológia teljes mértékben és viszonylag könnyen automatizálható: a hasábszerűen kiképzett, a két végén forrasztható kivezetés nélkül gyártott elemeket a nyomtatott áramköri lapon helyezik el, és az adott helyen forrasztással (esetleg ragasztással) rögzítik.

A készülékhez adott dokumentáció is más. Már nem szerepelnek kapcsolási rajzok, inkább csak blokkvázlatok, valamint az egyéb egységekhez csatlakozó pontok és bekötések elnevezései. Ennek az az oka, hogy az alkalmazott áramkörök nagy része már nem katalógus-áramkör, hanem egy-egy erre szakosodott cég (például: Chips and Technologies, Headland) terméke.

Forrasztópáka és XT eladó...

Halad a világ. Nem figyelték meg, hogy egy tévészerelő már nem annyira megszokott vendég? A televíziógyártásban alkalmazott elektronikai technológia — hasonlóan a számítógépgyártásához — igen nagy megbízhatóságot ad. Nagyon népszerűek az OSD (on screen display) technológiájú készülékek, amelyeknél a kezelés egy-két nyomógombbal vagy távszabályzóval elintézhető, és a funkciókat a képernyőn jelenítik meg. Ez nemcsak divat, hanem annak a felismerésnek az alkalmazása, hogy a mechanikus kezelőgombok, szabályozók könnyebben elromlanak. A legtöbb elektromechanikai rendszer azonban igen megbízhatóan működik.

Jelenleg egy elektronikus berendezés erkölcsi elavulása előbb következik be, mint annak fizikai tönkremenetele. Azt hiszem, most már érthető a szomorúságom. Vége a forrasztópákás korszaknak, a hibakereséssel, majd boldog reparálással megnyerhető sikereknek. Hiszen hogyan lehet egy felületszerelt technológiával készült berendezést javítani? Nagyon nehezen. Az igen kisméretű integrált áramköröket, ellenállásokat, kondenzátorokat sok esetben elektronsugárral forrasztják, tűhegyű páka használata nehézkes. Minimális kézremegéssel is zárlatot okozhatunk az IC-k lábain az oszcilloszkóp mérőszondájával. Meg aztán mit mérjünk? Kapcsolási rajz nincs, és még ha meg is találtuk a hibás alkatrészt, hogyan és mennyiért szerezzük be azt?

Ezért a javítások nagy része jelenleg kártyacsere. Tessék mérlegelni: a két soros és egy párhuzamos portot tartalmazó kártya ára áfával együtt 1250 Ft. Érdemes egy ilyen kártyát javítani? Természetesen nem.

Ez az imponáló fejlődés mégis pesszimisztikus jövőképet (is) fest elém. Eddig azért volt sok mérnökre, szakemberre szükség, mert a termelő berendezésekben alkalmazott technológia működtetése, javítása legtöbbször magas szintű, speciális szakértelmet igényelt. Az új berendezések ritkán meghibásodó részegységeinek cseréje már nem igényel ilyen tudást.

Jó példa az autók elektronizációja. A klasszikus autók elektromos berendezéseit az autovillamossági műszerész szakmunkás javította. Mit tesz ma egy ilyen szakember, ha valaki azzal viszi hozzá speciális nyugati kocsiját, hogy tönkrement az elektronikus gyújtás vagy más elektronikus egység? Egyet tehet: széttárja a karját, és olyan szervizbe küldi a tulajdonost, ahol speciális műszerek segítségével állapítható meg a hiba, de a javítás legtöbbször ilyenkor is a hibás egység cseréje.

Jobb sem félni, sem megijedni!

Ahhoz, hogy a legújabb bejelentésű chipből komplett, működő PC legyen, csupán memóriára, billentyűzetre és egy monitorra van szükség. A tok maga statikus működésű és kis fogyasztású, ezért ideális a zseb-PC számítógépekhez. A bejelentést úgy időzítették, hogy az egybeesett az IBM-PC tízéves jubileumával.

Már készülnek a szintén egy tokba integrált PC-k fejlettebb, 80x86-os processzort tartalmazó típusának a piacra dobására. A jövő ebből a nézőpontból nagyon izgalmas. Az áramköri szintű szemléletet és ismereteket egyre szűkebb területen lehet alkalmazni. Valószínű, hogy a mérnöki fejlesztő tevékenységet fel fogja váltani a rendszereket részegységekből összeállító, rendszerintegráló tevékenység. A biztos út a szakértelem átmentésére: a rendszerben való gondolkodás.

Dinamikus és statikus

A nagybonyolultságú digitális áramkörök — mint például a mikroprocesszorok, kalkulátor-áramkörök — belső működésükhöz ütemező jelet igényelnek, az órajelet. Ez az órajel biztosítja azt is, hogy az áramkör belső tárolói frissítéssel (folyamatos újraírással) megőrízzék tartalmukat. Ha ez az órajel megszűnik, a tok nem képes helyesen működni. Azt mondjuk, hogy a tok dinamikus működésű.

Az újabb fejlesztések eredményeként már gyártanak olyan mikroprocesszorokat (ezek a 80x86-os típusok), amelyeket leállítva — azaz órajelüket megszüntetve — majd újraindítva helyesen működnek. Az ilyen tokokat statikus működésűeknek nevezzük. A statikus működésnek köszönhetően álló állapotban a processzor fogyasztása igen alacsony, ezért a hordozható számítógépekben elsődlegesen az ilyen típusokat alkalmazzák.

Kónya László

Párhuzamosok, mégis találkoznak II.

Taszkkommunikációs eszközök

A párhuzamos programoknak, illetve futtatási környezetüknek — tágabban: a multitaszking rendszernek — a biztonsága és életképessége érdekében különböző kommunikációs protokollokat és átkapcsolási mechanizmusokat állítottak csatasorba. Ezeket mutatjuk be Modula-2 platformon.

A taszkkommunikációs eszközök nagymértékben elősegítik a logikai hibák elkerülését a program tervezése folyamán. Ezek az eszközök tulajdonképpen adatcsere-protokollok, valamint processzátkapcsolást vezérlő mechanizmusok (lásd a keretes részt).

Általuk biztosíthatjuk, hogy a párhuzamos program különböző processzei között meglegyen az összhang, és a program korrektül, a logikai tervezésnek megfelelően fusson; ne kerüljön deadlock állapotba, minden processz a szerepének megfelelő módon működjön, valamint hogy a processzek közötti adatforgalom logikailag hibátlan legyen.

Dijkstra és a szemaforok

Az absztrakt kommunikációs protokollok kifejlesztésének első nagy eredménye Dijkstra nevéhez fűződik. Ő volt az, aki a szemaforok mechanizmusát bevezette a multiprogramozásba. A szemaforokat könnyű implementálni, és elég erős eszközök ahhoz, hogy velük sok taszkkommunikációs problémát elegánsan megoldjunk. A szemaforok lehetővé teszik azt is, hogy erősebb és strukturáltabb kommunikációs eszközöket tervezzünk és implementáljunk.

Definíció szerint a szemafor egy egész típusú változó, amely csak pozitív értékeket vehet fel. A szemaforok két csoportba oszthatók. Amelyek csak két értéket (0 és 1) vehetnek fel, a bináris szemaforok; amelyek akármilyen pozitív egész számot felvehetnek, az általános szemaforok. Miután megadtuk a szemafor kezdőértékét, csak két művelet hajtható végre rajta. Az egyik a wait(sz) (várj sz-re), a másik a signal(sz) (küldj sz-t). A műveletek a következők:

wait(sz : Szemafor) → Ha sz > 0, akkor sz:=sz—1, ellenkező esetben annak a processznek a végrehajtása, amelyik meghívta a wait műveletet, felfüggesztésre kerül.

signal(sz : Szemafor) → Ha van olyan processz, melynek a végrehajtása megszakadt egy korábban kiadott wait(sz) hatására, akkor a vezérlés átkerül ehhez a processzhez, ellenkező esetben sz:=sz+1.

Csak a wait és a signal műveletek engedhetők meg a szemaforváltozókon; különösen tilos a szemaforváltozó értékének direkt írása vagy értékének vizsgálata. A wait és a signal műveletek alapműveletek ebben a szerkezetben, tehát végrehajtásuk egymást kizáró fo-

Multiprogramozást támogató eszközök Modula-2-ben

A program az algoritmus leírása, a processz az algoritmus végrehajtása. A program elindításának pillanatában létrejön a főprocessz, amely később maga is hozhat létre újabb processzeket. Ezek a processzek egymással párhuzamosan hajthatók végre. Ha a Modula rendszer egyprocesszoros gépre van telepítve, akkor egyszerre csak egy processz futhat, és a többinek várakoznia kell addig, amíg a vezérlés sora rájuk nem kerül. A szabvány Modulában a következő utasítások állnak rendelkezésre ahhoz, hogy új processzeket hozzunk létre, és a processzek között átadjuk a vezérlést.

```
NEWPROCESS(proc:PROC;cim:ADDRESS;méret:CARDINAL;VAR
task:PROCESS)
```

Ez az eljárás létrehoz egy processzt, amely a paraméter nélküli proc eljárásból áll. A processznek szüksége van egy munkaterületre, amelynek címét a cím paraméter, méretét pedig a méret paraméter adja meg. Ha a munkaterület túl kicsi, akkor a program futásideji hibával leáll. TopSpeed Modulában a munkaterületnek minimum 1000 bájtnak kell lennie. Ha sikerült az új processz létrehozása, akkor a task változóban kapjuk vissza azt a kezelőt, amellyel ezután hivatkozhatunk a processzre.

```
TRANSFER(VARmostani,következő:PROCESS)
```

Ha át akarjuk adni a vezérlést egy másik processznek, akkor ezt az eljárást kell meghívnunk. Az eljárás meghívása után az éppen futó processz azonosítója bekerül a mostani paraméterbe, és a processz végrehajtása felfüggesztődik, a vezérlés a következő paraméter által azonosított processzre kerül.

```
IOTRANSFER(VARmostani,következő:PROCESS;vektor:
CARDINAL)
```

A processzek közti átkapcsolás történhet tudatosan egy másik processz kezdeményezésére, de történhet véletlenül is, amikor egy bizonyos megszakítás hatására kapja meg a vezérlést egy bizonyos processz. Ahhoz, hogy olyan processzünk legyen, amely kezeli a megszakítást, először a NEWPROCESS eljárással létre kell hozni, majd egy TRANSFER utasítással át kell neki adni a vezérlést. A processz ezután végrehajthat néhány műveletet, majd ciklusba kerül. A ciklusban az első dolga az, hogy meghívja az IOTRANSFER eljárást. Ez az eljárás hasonlóan működik, mint a TRANSFER, de definiál egy megszakító vektort is. Ha az adott megszakítás létrejön, akkor az éppen futó processz felfüggesztésre kerül, és az IOTRANSFER-t meghívó processz kapja meg a vezérlést.

Azért, hogy elkerüljük a megszakítás újbóli generálását, amíg a rendszer a megszakításkezelőt hajtja végre, a modulnak, amely a kezelő processzt deklarálta, prioritást kell adni. Ezt a prioritást közvetlenül a modul azonosítója után, szögletes zárójelben adhatjuk meg. A prioritás egy CARDINAL típusú szám, amely miután BITSET típussá konvertálódik, logikai összeget képez a megszakítást kezelő áramkör megszakító maszkjával. Ez a mechanizmus azt eredményezi, hogy amíg a program a modul által deklarált eljárásokat hajtja végre, addig a megszakításvezérlő megszakító maszkja azt az értéket veszi fel, amely az eredeti érték és a modul prioritásának logikai összege. E mechanizmus által tehát letilthatunk különböző hardvermegszakításokat.

lyamat. Tilos egyszerre több processzben ezeket a műveleteket egyszerre végrehajtani. A signal művelet nem köti ki, hogy melyik processzt kell reaktiválni, ha több processz vár ugyanarra a szemaforra.

Ezért a szemafor implementációjától függ, hogy milyen sorrendben kapják meg a vezérlést azok a processzek, amelyek ugyanarra a szemaforra várnak.

Az OOP és a monitorok

Az olyan operációs rendszer, amely csak szemaforokat használ, nagyon sérülékeny: elég, ha csak egy szemaforművelet van rossz helyen, vagy egyáltalán hiányzik valahonnan, máris az egész logikai rendszer felborulhat. Szükséges tehát egy olyan eszköz — a monitor —, amely saját maga ügyel rá, hogy a kritikus területhez egyszerre csak egy processz férjen hozzá. A monitorok ötlete két szálon érett be.

Az egyik bejáratott gyakorlat: az operációs rendszerek funkció-végrehajtó magjáé. A legtöbb ilyen mag egy nagy program, amely minden operációsrendszer-függvényt kontrollál, és csak ő hajthatja végre. Hogyha a P1 processz üzenetet akar küldeni a P2 processznek, akkor az üzenetet elküldi az M monitornak azzal az utasítással, hogy ez az üzenet a P2 processzhez tartozik. A monitor elintézi az üzenet átadását. Fő funkciója még az erőforrások feletti abszolút felügyelet. M működését nem lehet megszakítani, ami biztosítja a kölcsönös kizárás elvének betartását. Csak a monitor tud hozzáférni különböző speciális memóriaterületekhez, valamint csak ő tud végrehajtani bizonyos műveleteket, például az I/O műveleteket. (Azok a monitorok, amelyekkel mi foglalkozunk, az operációsrendszer-monitorok decentralizált verziói lesznek. Ezekből lehet egyszerre több is a rendszerben, és mindegyik egy bizonyos funkciót lát el.)

A másik tényező az adatok és a hozzájuk tartozó műveletek strukturalizációjára való törekvés. Ebből nőtt ki a 90-es évek programtechnikai forradalma, az objektumorientált programozási stílus. A monitor tehát lehet (és meg is valósítható mint) egy objektum, amelyben az adatok és az adatokhoz tartozó műveletek együttesére igaz, hogy csak a monitor tud hozzáférni a monitoron belül definiált adatokhoz, továbbá a monitor és a külvilág kapcsolata csakis a monitor publikus procedúráinak paraméterein keresztül zajlik.

Ha nem állnak rendelkezésre objektumorientált eszközök, akkor a monitor megvalósítható különálló könyvtár formájában, amely csak a külvilággal kapcsolatot teremtő procedúráit „nyitja ki", és az adatokat, valamint a privát eljárásait csak az implementációs részben tartalmazza. A monitorokat könnyű implementálni a szemaforok segítségével, és ez lehetővé teszi, hogy a monitorban levő eljárások tényleg megszakítás nélkül futhassanak, egyszerre csak egy processz hívására — ez elengedhetelen a kölcsönös kizárás elvének betarthatásához.

Ada és a randevú

A szemaforok és a monitorok centralizált szinkronizációs eszközök, amelyek nem tartoznak egyetlen processzhez sem. Saját maguk szabályozzák a processzek közti átkapcsolás sorrendjét és rendszerét. Ez a szerkezet nagyon jól megfelel egy olyan rendszerben, amelyben csak egy számítógép van, de mi a helyzet az osztott rendszerek esetében? (Azokat a rendszereket nevezzük osztott rendszereknek, amelyek több egymástól független számítógépen futnak, és a rendszer részei közötti egyetlen kapcsolat kizárólag üzenetek cseréjén alapul.) Ez viszont a következő problémát veti fel: az üzenetek kiküldése és felvétele között nincs semmi szinkronizáció, az üzenetek elveszhetnek útközben, és a rendszert alkotó számítógépek közül is meghibásodhat néhány példány. Ilyen helyzetben egy olyan önszabályzó protokollra van szükség, ahol a processz saját maga dönt arról, hogy vár-e (és meddig) az üzenetre. Ha a processz úgy dönt, hogy vár, akkor saját maga függeszti fel a működését, és saját maga dönt arról, hogy meddig fog várni. (A szemaforok és a monitorok esetében más volt a rendszer: ahhoz, hogy egy processz reaktivizálódjon, egy másik processznek ki kellett küldenie egy szignált.) Egy ilyen protokollnak egyetlen fontos eleme a processzek közötti kommunikációs csatornák biztosítása, és az üzenetek kicserélése e csatornák felhasználásával. Ezt az üzenetváltást nevezzük randevúnak.

A randevút Hoare ajánlotta 1978-ban „Cooperating sequential processes" című munkájában. Az eredeti verzió helyett azonban inkább az Ada nyelvben realizáltról ejtünk néhány szót. (Az Adáról lapunkban már több írás jelent meg. — A szerk.) A nyelv születésekor úgy hitték, hogy megalkották a végső nyelvet, amely véget vet az ún. szoftverkrízisnek. (Ez az a jelenség, miszerint a szoftver fejlesztési idejének csak egy egészen kis részét teszi ki magának a rendszernek a tervezése és implementálása, az idő túlnyomó részét a rendszer tesztelése teszi ki. Ha a rendszer meghibásodik, akkor a hibajavításra megint rengeteg időt kell szánni.) Később kiderült, hogy az Ada sem old meg minden problémát...

Az Adában a taszk egy objektumhoz hasonló típus. A randevút három nyelvi elem biztosítja. Az első egy speciális típus, az ENTRY, amely hasonlít a procedúradeklarációhoz abból a szempontból, hogy meg kell adni az input és output paramétereket. E paraméterek kellenek az adatforgalomhoz a randevút lebonyolító két taszk között. A következő nyelvi elem a SELECT szerkezet, amellyel a taszk egy randevúkérelmet ad ki egy másik taszk ENRTY típusú belépési pontjára, és ha létrejön a randevú, akkor a SELECT szerkezetben definiált változók segítségével megy az adatcsere a taszkok között. A harmadik szerkezet az ACCEPT: az adott taszk ezzel jelzi, hogy kész elfogadni egy randevúhívási kérelmet egy adott ENTRY típusú belépési pontra. Mind a SELECT, mind a ACCEPT szerkezetben definiálhatunk alternatív műveleteket, amelyeket akkor hajt végre a taszk, ha valamiért nincs randevú. (Akit komolyabban érdekel az Ada nyelv és a taszkrandevú, olvassa el I. C. Plyde „Az Ada programozási nyelv" című könyvét.)

A TopSpeed Modula-2 multitask rendszere

Az alapkönyvtárak között van a multitaszk programozást segítő könyvtár is. Ez a könyvtár a következőket biztosítja számunkra:
- általános szemafortípus, valamint a szemaforokon végezhető műveletek;
- egy dinamikus, egyenlő időszeletű időosztáson alapuló átkapcsoló mechanizmus;
- maximum 32 egyszerre futó processz;
- maximum 65 534 szintű prioritáshierarchia;
- futásidejű prioritásszint-változtatás;
- direkt taszkátkapcsolási lehetőség;
- kritikus területek kezdetének és végének jelzésére szolgáló lock/ unlock funkció;
- teljes körű együttműködés a Windows könyvtárral.

A lemezmellékleten van egy könyvtár, amely az Ada taszkrandevú funkcióját implementálja bizonyos megkötésekkel, Modula-2-ben. A könyvtár néhány lehetőségét demonstrálja egy példaprogram, amely egy klasszikus multiprogramozási feladatot old meg a randevú segítségével. Ez az ún. „öt filozófus probléma", melyben a multiprogramozás szinte minden buktatója benne van.

Öt filozófus egész nap nem csinál mást, mint új filozófiai elméleteket gyárt, és mindegyik örülne, ha csak a gondolkodással kellene foglalkoznia. De időnként megéheznek, és ezért átmennek a konyhába, ahol egy nagy közös asztalnál mindenkinek van egy tányérja és egy villája. Az asztal közepén egy soha ki nem ürülő spagettis tál van. Sajnos a filozófusok csak két villával tudnak étkezni, ezért ha valamelyikük enni akar, akkor a saját villáján kívül szüksége van a hozzá legközelebb eső másik villára is. Tehát egyszerre csak kettő lakhat jól az asztalnál, és a többinek várakoznia kell, amíg nem lesz szabad villa. Egyezkedni muszáj... (A probléma lényege tulajdonképpen nem más, mint az erőforrások egyenlő, igazságos elosztása az igénylők között.)

Dobi Sándor

A C nyelv Unix-felülete III.

Olvasás innen-onnan, így és amúgy

Nemcsak írni, olvasni is tudni kell! A múlt hónapban volt szó a kiíró utasításokról — szóval a felét már megtanulhattuk a dolgoknak —, most jön a bonyolultabb rész...

A scanf, fscanf, sscanf utasítások és a read elsődleges bemutatása után a mélyebben rejlő tudnivalókról is szó lesz, és ennek kapcsán más fontos fogalmakról, technikákról adunk ismertetést.

Az utasításhármas

A scanf utasítással a standard bemenetről, míg az fscanf utasítással egy fájlpointeren keresztül fájlból, az sscanf utasítással pedig memóriából lehet ASCII formátumú adatokat olvasni és átkonvertálni.

Az utasítások szintaxisa a következőképpen fest:

```
scanf(format[,pointer,
pointer...]);
char *format;
fscanf(fp,format[,pointer,
[pointer...]);
FILE *fp;
char *format;
sscanf(s,format[,pointer,
[pointer...]);
char *s, *format;
```

A 'format' vezérlőstring a legegyszerűbb esetben csak konverziós specifikációkat tartalmaz. Egy konverzióspecifikáció egy %-jelet követő valamilyen karakterből áll. A megfelelő argumentum (egy mutató) azt adja meg, hogy hová kell konvertálás után betölteni az adatot.

A leggyakoribb típusok a következők:

```
d (decimális egész)
e,f,g (fix vagy
lebegőpontos float)
s (string)
```

A bemeneten a fehér karaktereknek (szóköz, tabulátor, új sor) delimiter szerepük van, a formátumstringben specifikált mezők elválasztására szolgálnak. (Kivétel a későbbiekben ismertetendő karakter- és karakterosztályt-ípus.) A stringeket (%s) is úgy kell értelmezni, hogy fehér karakterek által határolt karaktersorozatok. String olvasásakor a memóriába bekerül a lezáró 0 bájt is. Például:

```
fp = fopen("DATA","r");
fscanf(fp,"%d%d",&x,&y);
printf("x=%d y=%d\n",x,y);
fscanf(fp,"%f%e%s",&f1,
&f2,buff);
printf("f1=%f f2=%f
buff=%s\n",f1,f2,buff);
```

DATA tartalma:

```
   11 22
      1.234E05 5.678
 ABCD
```

Kimenet:

```
x=11 y=22
f1=123400.000000
 f2=5.678000 buff=ABCD
```

Gyakori hiba kezdőknél, hogy az argumentummegadásnál mutató helyett magát a változót adják meg (ez rossz!):

```
fscanf(fp,"%f",f1);
```

A karakter- és karakterosztály-típus kivételével (ezekről később lesz szó) a konverzióspecifikációk között elhelyezett szóközöknek nincs jelentősége:

```
fscanf("%f%f",&f1,&f2) =
= fscanf("%f %f",&f1,&f2)
```

Puffer nélkül

A read a blokkolt, puffereletlen olvasás előírása. Fájldeszkriptorral megnyitott fájlból a read utasítással lehet (puffereletlenül) adott számú bájtot olvasni:

```
read(fd,buf,nbytes);
int fd;
char *buf;
unsigned int nbytes;
```

Itt fd a megnyitott fájlra mutató deszkriptor, buf a memóriapuffer címe, ahová a beolvasott karakterek kerülnek, nbytes a beolvasandó bájtok száma. A read visszatérő értéke a ténylegesen beolvasott bájtok számát adja meg sikeres művelet esetén, különben —1. (Ha például fd nem egy nyitott fájlra mutat, vagy ha a hozzáférési jogok miatt nem olvasható a fájl.) Például:

```
fd = open("DATA",0);
n = read(fd,buff,2);
printf("n=%d buff[0]=
%c buff[1]=
%c\n",n,buff[0],buff[1]);
n = read(fd,buff,2);
printf("n=%d buff[0]=%c
buff[1]=%c\n",n,buff[0],
buff[1]);
```

DATA tartalma:

```
ABC
```

(3 karakter volt a fájlban, tehát a végén nem volt újsor-karakter!)

Kimenet:

```
n=2 buff[0]=A buff[1]=B
n=1 buff[0]=C buff[1]=B
```

A write a blokkolt, puffereletlen írás kezdeményezése. Fájldeszkriptorral megnyitott fájlba a write utasítással lehet (puffereletlenül) adott számú bájtot írni:

```
write(fd,buf,nbytes);int fd;
char *buf;
unsigned int nbytes;
```

Itt fd a megnyitott fájlra mutató deszkriptor, buf a memóriapuffer címe, ahonnan a karaktereket ki kell másolni, nbytes a kiírandó bájtok száma. A write visszatérő értéke a kiírt bájtok számát adja meg sikeres művelet esetén, különben —1. (Ha például fd nem egy nyitott fájlra mutat, vagy ha a hozzáférési jogok miatt nem írható a fájl.)

Például:

```
fd = open("DATA",1);
strcpy(buff,"12345");
write(fd,buff,3);
```

DATA tartalma a kiírás előtt:

```
abcdefg
```

DATA tartalma a kiírás után:

```
123defg
```

Nemes Mihály

Ért Ön Snobolul?

Ne kezdjük újra!

Mostani számunk mágneslemez mellékletén olyan nyelv fordító-értelmező programját adjuk közre, amely után sokáig hiába ácsingóztak nagy európai számítóközpontok munkatársai is. Kérjük olvasóinkat, hogy jól őrizzék meg; lehetőleg készítsenek egy külön könyvtárat, mert szükségük lehet rá az elkövetkezendő hónapokban. Cikksorozatot indítunk a Snobol nyelv megismertetésére, és a mágneslemez mellékleten programozási fogásokat mutatunk be, hogy kedvet csináljunk a nyelvben rejlő lehetőségek kipróbálására.

Tudomásom szerint Európában hosszú ideig egyedül a hollandiai Matematikai és Számítástudományi Központ birtokában volt belőle egy példány, mások csak könyvekből értesülhettek a nyelv (és a felhasználását biztosító fordító-értelmező program) imponáló megoldásairól.

Többek között azokból a tanulmányokból és kutatási jelentésekből is, amelyeket éppen a szóban forgó amszterdami központ munkatársai publikálták a nyelv látszólagos egyszerűségéről és rejtett bonyolultságáról, valamint a felbukkant új gondolatok továbbfejlesztési lehetőségeiről.

Miért nincs, ha van?

Valódi unikum, hogy egy ilyen nagy hírű programrendszer teljes verziója, ha méretkorlátozásokkal is, de a programozók népes táborának közkincsévé válhatott (lásd a 92/6. szám Közkincs rovatát).

A nyelv a Bell Laboratóriumból indult el kontinenshódító útjára. (Amerikában a legtöbb nagy számítóközpontban már jó ideje hozzá lehetett férni valamelyik változatához.) Különösen az gyorsította meg elterjedését, hogy R. E. Griswold, a nyelv egyik megteremtője — aki később a C nyelv szakértőjeként vált közismertté — kifejlesztett egy önmagában is figyelemre méltó, speciális gépfüggetlen makroimplementáló nyelvet. Az ún. SIL makrónyelv (Snobol Implementation Language) nagyban megkönnyítette az implementáció elkészítését a legkülönbözőbb géptípusokra, és egyben biztosította az így készült szoftvereszköz hatékonyságát is.

Az első működő Snobol rendszerek IBM gépekre készültek, ezeket követte a Burroughs, az RCA, a CDC, a UNIVAC, az SDS, majd a General Electric és a PDP — mint látható, lényegében valamennyi nagy amerikai számítógépgyártó cég gépére készült fordító-értelmező. Mivel azonban ezeknek a kifejlesztését legtöbbször nem maguk a gyártó cégek finanszírozták, így terjesztésében sem voltak érdekeltek. Talán elsősorban ez magyarázza, hogy Európába alig szivárgott át. Nem tartozott bele a gyártó cégek standard kínálatába, és beszerzését is csak vonakodva vállalták, horribilis összegekért.

A nyelv egyik ismert alkalmazása arra keres megoldást, hogy milyen módszerekkel lehet közönséges szövegekből összegyűjteni, és formalizált alakra hozni a bennük rejlő hasznos információkat. Tudjuk, hogy bizonyos adatbázisok készítőinek a legnagyobb gondot a hatalmas mennyiségű és állandó mozgásban, változásban lévő kémiai információtömeg jelenti. Nos, elsősorban az ő dolgukat könnyíti az ún. SIE rendszer (Specialized Information Extraction System). Ez a szakmai információkivonatoló rendszer morfológiai, szintaktikai, valamint lexikológiai és szemantikai vizsgálatok alapján „nyers" angol nyelvű leírásokból tudja előállítani különböző kémiai reakciók tömör értelmező kivonatát.

Makrók — mikróban

A 80-as évek végén néhány bősz matematikusnak az az istentől elrugaszkodott gondolata támadt, hogy a makróimplementáló nyelv hatékonyságát megpróbálja mikroszámítógépeken is kamatoztatni. A kísérlet a várakozásnál is fényesebben sikerült. Kitűnően oldották meg a számítógépes rendszer memóriagazdálkodását; munka közben automatikusan folyik a háttérben a „szemétgyűjtés", a felszabadult hulladékmemória újrahasznosítása, a felhasználónak erre a legcsekélyebb figyelmet sem kell fordítania.

Az is hozzájárult a sikerhez, hogy előzőleg már sokszorosan tesztelték a nagygépeken, így az implementálóknak csak arra kellett kínosan ügyelniük, hogy a PC-kre készült változat az utolsó csavarig pontosan megfeleljen a nagygépeken futó mintarendszerek előírásainak. És tegyük hozzá, kezére játszott a PC-s minirendszerek készítőinek, hogy a hatékonyság már távolról sem kell, hogy annyi fejtörést okozzon, mint elődeiknek a nagygépeken...

A Griswold-féle makrónyelvre alapozott implementációk további előnye, hogy egészen kis terjedelműre szorítható össze a program tulajdonképpeni „magja". A bonyolultabb dolgokat már le lehet írni magában a Snobol nyelvben, például külső függvények formájában. Aki megnézi a mágneslemezen közölt SNO.COM program méretét, csodálkozva állapíthatja meg, hogy feltűnően kicsi. Snobol nyelven van írva már az interaktív használatot megkönnyítő CODE.SNO program is, amely egyébként saját létével egyrészt kézzelfogható bizonyítékát szolgáltatja a rendszermag bővíthetőségének, másrészt példát is mutat, hogyan lehet ezt megtenni.

Most akkor compiler vagy interpreter?

Logikátlannak tűnik a „fordító-értelmező" összetétel? Általános fogalmaink szerint vagy ez, vagy az, hiszen ez a két dolog elég jelentős mértékben eltér egymástól. A fordító futtatható kódot készít, amely csak később kel életre, futtatáskor, ha különböző bemenő adatokkal „tápláljuk". Az értelmezőről közismert, hogy nem különíti el egymástól a kódkészítés és a futás folyamatát, sőt a bemenő adatok és a környezet ismerete is annyira hozzátartozik magához az értelmezéshez, hogy nélkülük az el sem kezdődhet.

Hogy sarkosan fogalmazzunk: az értelmező lassú és buta, hiszen mindig mindent újra kezd. Nemcsak hogy valamennyi futás alkalmával, hanem akár még egyetlen futáson belül is akárhányszor előfordulhat, hogy minden apró tennivalót (még a később fölöslegeseknek bizonyultakat is) unos-untalan újból fel kell ismernie, és szolgaian végre kell hajtania. Az ilyen rendszer szervi hibája, hogy nem lát tovább az orránál — mondhatja valaki nem is alaptalanul. Vagy mégsem?

Nem lehet elvitatni, hogy a maga helyén az értelmező körülményességének is megvan a létjogosultsága. És nem is olyan biztos, hogy mindig igazságos ez a merev szétválasztás. Nem indokolt már a LISP-nél sem, és még sokkal kevésbé a Snobol esetében.

Jók-e a skatulyák?

A LISP nyelv implementációinál sem egészen a hagyományos értelemben beszélnek fordítóprogramról és értelmezőről. Nézzük előbb az értelmezőt. A LISP-implementációk „közönségesebb" változata maga is egy ciklikus üzemmódú interpreter, amely párbeszédet folytat a programozóval. Először türelmesen várja a feldolgozandó kifejezés bevitelét, majd mikor a „kész" jelet megkapja, a kifejezést kiértékeli. A harmadik fázisban kiírja a kiértékelés eredményét, azután újra kezdi a türelmes várakozást.

Ami a fordítóprogramot illeti, az a LISP értelmezője számára nem más, mint egy beépített függvény, amelyet alkalmazni lehet valamilyen nem beépített függvényre. Az alkalmazás következtében — tulajdonképpen a kiértékelés mellékhatásaként — jön létre az a bizonyos gépi kód, amelyre vártunk: az argumentumként megadott függvénynek a számítógép nyelvére lefordított változata. Alkalmazáskor ez az előre elkészített, gépi kódra lefordított függvény láthatatlanul helyettesíti azt a kifejezést, amely egyébként újra és újra értelmezésre szorulna. A fordítás és az értelmezés itt nincs szembeállítva: amaz kiegészíti és „rövidre zárja" ezt. A végrehajtás így jelentősen felgyorsítható, éppen azáltal, hogy a lefordított függvények, eljárások szervesen beleépíthetők az interpretálás folyamatába — ez szélsőséges esetben egyszerű futtatássá is korcsosulhat.

Ezáltal tehát jelentősen csökkenthetők az interpreterek hátrányai, megmaradnak viszont az előnyei: az egyszerű programfejlesztés, az azonnali kipróbálhatóság, a módosítások könnyű megvalósítása.

Bizonyos szempontból analóg a helyzet a Snobollal is, hiszen interpreterének is van olyan beépített függvénye, amely hasonló szerepet tölt be, mint a COMPILE függvény a LISP-ben. Itt azonban olyasmiről is szó van, ami miatt a Snobol „interpreterét" funkcionálisan meg kell különböztetnünk a hagyományos interpreterektől. Pontosan ezt kívántuk érzékeltetni, amikor a „fordító-értelmező" elnevezést használtuk.

A Snobol interpreterének tevékenysége szétválik két szakaszra: „szintaktikázásra" és „szemantikázásra". Az első szakaszban ez az okos kis jószág végigmegy a Snobol nyelven írt programon, de a program előírásainak valóságos végrehajtása nélkül. Csak úgy mellékesen viszont elvégzi a program szintaktikus elemzését. Sőt fordít is: lefordítja a programot egy gépfüggetlen belső kódra. Ez nyilvánvalóan azzal jár együtt, hogy közben feltárja a programot alkotó elemek egymáshoz kapcsolódásának belső szerkezetét is. És természetesen elküldi a felhasználónak a megfelelő hibajelzéseket. Ehhez megjeleníti a képernyőn a forrásprogram hibás részét, sőt lokalizálja a hiba jelentkezésének a helyét, és szövegesen kiírja a hiba típusát is.

A második szakaszra fatális hiba esetén már nem is kerül sor mindaddig, míg a programozó nem javít. Kisebb hibák esetén a rendszer azonnal megpróbálkozik a futtatással, hogy ezáltal több információ álljon rendelkezésre a hiba kijavításához. Azt talán már felesleges is hozzátenni, hogy a „szemantikázás" nyomán feltárt hiba jellegéről és előfordulási helyéről is tájékoztatást kap a programozó.

A kétszakaszos megoldás jelentős mértékben növeli a Snobol rendszer hatékonyságát: az első szakasz végeredményeként már kész egyrészt egy szintaktikailag hibamentes forrásprogramrészlet, másrészt ennek egy olyan belső kódú ábrázolása, amely már sokkal könnyebben és gyorsabban végrehajtható. (Talán nem is olyan biztos, hogy minden interpreter buta...)

Vargha Dénes

(Cikkünk a 14. oldalon.)

INFORMÁCIÓKÉRÉS: 31 ▶

MAXELL FLOPPYLEMEZEK NYÁRI VÁSÁRA!!

	1–10 doboz	11–50 doboz	51 doboz felett
	(Árak: Ft/doboz)		
MAXELL 5.25" DS/DD	700	670	640
MAXELL 5.25" DS/HD	990	960	920
MAXELL 3.5" HD	1800	1700	1650

Földelhető polarizált üvegszűrők bevezető áron!

1–10 db	11–50 db	51 db felett
2250 Ft/db	1750 Ft/db	1450 Ft/db

ÁRAINK A 25% ÁFA-T MÁR TARTALMAZZÁK!



UNITRADE

Szervezési, kereskedelmi és Számítástechnikai

K.F.T.

1073 Budapest VII., Erzsébet krt. 48.
Telefon/Fax: 142-2115

...nemcsak számítástechnika

INFORMÁCIÓKÉRÉS: 33 ◀



TONER KFT

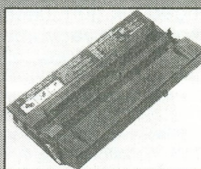


Másológép és lézerprinter-kazetták felújítása német technológia alapján, garanciával, feketében és színesben.

Europa International

NE DOBJA EL!

Felújítható kazettatípusok: FC, PC, EP, EP-S, EP-L, SHARP Z-30, Z-50.

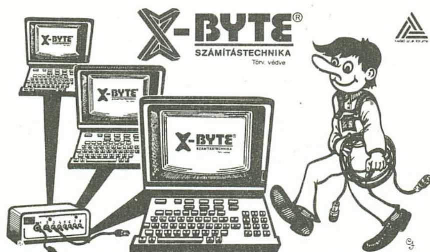


1095 Budapest Mester u. 21. Tel.: 113-1687 Tel./Fax: 134-3516

INFORMÁCIÓKÉRÉS: 32 ▶

X-BYTE®

SZÁMÍTÁSTECHNIKA
Törv. védve



KAPCSOLÓDJON A JÖVŐHÖZ!

SZÁMÍTÓGÉPHÁLÓZATOK

MILYEN TÍPUSÚ HÁLÓZAT SZÜKSÉGES ÖNNEK?

ARCNET, ETHERNET, RS 232,
IBM CABLING SYSTEM AT & T,
SYSTIMAX, ÜVEGSZÁL,
NOVELL?

JÖJJÖN EL HOZZÁNK!

1138 Budapest, Népfürdő u. 17/e. Telefon: 173-1329 Fax: 173-1530

Egy kávé és üdítő mellett segítünk a választásban.

CSÖKKENTETT ÁRAK, VÁLTOZATLAN MINŐSÉG!

INFORMÁCIÓKÉRÉS: 34 ◀

HC Hun Comp

AKCIÓ!! AKCIÓ!!

■ DATAS SVGA (1024X7/8) monitor	25 000,– Ft + ÁFA
■ AT 386/40 MHz számítógép 1.2 MB FDD-vel baby házban: (monitor és wincsi nélkül, egyébként kompletten)	49 500,– Ft + ÁFA
■ 40 MB-os wincsi	19 000,– Ft + ÁFA
■ 120-as wincsi	37 000,– Ft + ÁFA
■ NOVEL 3.11 20 user:	199 000,– Ft + ÁFA

AKCIÓ!! AKCIÓ!!

HunComp

1116 Bp. Mohai út 37. Tel & FAX: 185-4186

Számkígyó-bűvölés

Januári agytorna-gyakorlatunk, akár a decemberi, komoly kihívásnak ígérkezett. Egyetlen kifejezés, egyetlen szám vagy számjegy előzetes ismerete nélkül kellett a megadott aritmetikai összefüggésekből meghatározni egy ismeretlen nyelv számrendszerének összes csínját-bínját. Ráadásul még egy ijesztgetésnek is beillő figyelmeztetés vezette be a feladványt: hogy nem mindig előny, ha engedünk gondolkodásunk korábbi beidegződéseinek, és képtelenek vagyunk szabadulni „egyedül lehetségesnek képzelt gondolati formáinktól".

Nosza, több sem kellett a gondolati formáktól szabadulni vágyó, szárnyalásra kész programozói fantáziának. Jó néhányan nemcsak az alapfeladatot és az extrafeladatot oldották meg elismerésre méltó találékonysággal, hanem még „szorgalmit" is beküldtek. Többen „oda-vissza" programokat készítettek, amelyek az ún. arab számjegyekkel felírt számokat lefordítják az ismeretlen nyelvre (most már elárulhatjuk: a kínaira), illetve megfordítva: a sárkánykígyó hosszúságú kínai számnevekből elővarázsolják „közönséges", számjegyekkel felírt alakját.

A kínaiak saját számneveikkel írják a számokat — persze a számnevekhez tartozó hieroglifáikkal jelölve. Ma már terjed ugyan (főleg Tajvanon) az „arab" számjegyírás, de a hagyományaikra büszke kínaiak még a lapszámozásban, és az évszámok jelölésében is máig ragaszkodnak szokásaikhoz.

Az osztás mint egyetemi tananyag

Ha már itt tartunk, szenteljünk egy pár sort a voltaképpen csak nevében arab számjegyírásunk történetének. Ami „arab", az egyrészt eredetét tekintve indiai, másrészt a mai arabok nem is ezt, hanem ennek egy igen-igen távoli rokonságot mutató változatát használják.

Viszont Európába arab közvetítéssel jutott el az indiai számírás.

Először Abu Abdallah Muhammed ben Musza al-Hvarizmi (aha! az algoritmus akaratlan névadója) írt könyvet az indiai számírási módról és az ehhez kötődő műveleti szabályokról. Latin fordítása („De Numero Indorum") tette lehetővé, hogy lassacskán Európában is megismerjék a keleti számolóművészet titkát.

A bagdadi könyvtár rejtett kincseit feltáró matematikus, a horezmi születésű Mohamed érdeme múlhatatlan a numerikus számolás fejlődésében: a nyugati világban az ő munkásságának eredményeként terjedt el a helyi értékes, nullát is jelölő tízes számrendszer. Egyébként a „horezmi" maga még aligalig használt szimbólumokat, a számokat is inkább kiírta betűkkel. A IX-X. században mégis a könyvből ismerték meg és kezdték tanítani az „új módit" a spanyolországi arab egyetemeken. A szabályokkal jól megfoghatóvá tett osztásnak például a mór kereskedők is hasznát vették, és a formalizálás aligha sikerülhetett volna a római számok alapján. A római számok technikájával 5000-től leírni sem tudjuk a számokat!

Spanyolországi egyetemeken ismerte meg az osztás tudományát a francia Gerbert szerzetes is, aki később II. Szilveszter pápaként vonult be a történelembe. (Aha! Szent István koronája.) Meg is vádolták: nyilván az ördöggel cimborál, hiszen hogy lenne képes másképp bármekkora számokat elosztani egymással... Pápaként sem tudta eloszlatni a hagyományos jelölés híveinek ellenkezését. Hosszú, évszázadokig elhúzódó harc kezdődött a szabadelvű „algoritmikusok" és a konzervatív „abakuszosok" között, és a XIII-XIV. századig inkább az utóbbiak voltak fölényben.

A fordulatot egy pisai fiatalember, Leonardo Pisano készítette elő a XIII. század elején. Műve, a „Liber Abaci", vagyis az Abakuszok Könyve nem az abakuszok használatát, hanem éppen ellenkezőleg, szükségtelenségüket magyarázta el nagyon világosan. Leonardo, a Bonaccio-fiú — vagyis Fibonacci (van-e számítástechnikus, ki e nevet nem ismeri?) — számos gyakorlati példán mutatta meg, mennyivel könnyebb helyi értékes számokkal számolni, mint a hagyományos módon.

Számok — verselve és énekelve

A igen nagy és hosszú számok kiszámításában és megtanulásában is az indusok jártak az élen. Csillagászati könyveikben egész szinusztáblázatokat foglaltak versbe. Írásos hagyományaik szerint a tíz 140-edik hatványára is volt szavuk (asamkhyeya). Széles körben használt rendszerről azonban nem beszélhetünk, hiszen például a nagy számok megnevezését illetően nem egységesek a különböző forrásművek. Az iskolák mindegyike a maga hagyományait ápolta. Tény viszont, hogy a számok potenciális végtelensége és a nagy számok közötti rendcsinálás régóta izgatta az indiai matematikusokat. Bizonyára ez vezetett ahhoz, hogy voltaképpen rájöttek a rekurzív definíció lehetőségére.

A ma élő nyelvek közül a kínaiban található „természetes nyelvi eszköz", tehát külön tőszó a legnagyobb természetes szám kifejezésére. Ez a legnagyobb szám a százmillió, vagyis a tíz 8-adik hatványa. A nagy számok tagolása eltér a mienktől: nem hármasával (illetve hatosával) tagolják a számjegyeket a kínaiak, hanem négyesével. A százmillió is azért lehet tőszó, mert tízezerszer tízezer.

Ezt a számot írtuk feladatunkban Yü-nek, hogy világosan megkülönböztessük az '1' jelentésű Yi-től. Valójában ez is Yi, csak „másképp éneklik". Zeneileg egyvonalas c-vel jelölhetjük az '1' jelentésű Yi-t, amelynek tónusa kitartott (1. tónus), a '100.000.000' jelentésű Yi viszont ereszkedő tónusú (4. tónus), amely ugyanerről a hangról indul, de kis szext lépéssel leereszkedik a kis e-re (szolmizálva: mi-do).

Kegyetlenül nehéz hangtani átírásban visszaadni a kínai nyelv kiejtését. Minden magánhangzónak négyféle tónusa lehet: kitartott, emelkedő, ereszkedő-emelkedő és ereszkedő; a tónustól függően más-más lehet a szavak jelentése. A „ma" szó négyféle jelentése: 1. anya, 2. kender, 3. ló, 4. káromkodni. (Micsoda táptalaja a félreértésnek!)

De ez még mind semmi. Mert ha a tónusokat pontosan megkülönböztetjük is, akkor is marad rengeteg azonos

hangzású szó, amelyek kiejtésükben nem térnek el egymástól, csupán a hieroglifájuk különböző. Például a Yi szónak az első és a második tónussal 7-7 különböző hieroglifa felel meg, a harmadikkal 5, a negyedikkel — nem tévedés — 21! Nem csupán hagyománytisztelet vezeti hát a kínaiakat, amikor nem egykönnyen mondanak le megszokott írásmódjukról.

Kínai számok kódolása és dekódolása

„A bevezető alapján a legvadabb dolgokra gondoltam: nem tízes számrendszer, római számokhoz hasonló kivonásos számképzés..." — írja Csurgay Péter Csepelről.

Legprecízebb megfejtőnk, Dezső András is fel volt készülve minden rosszra: „Szinte biztos voltam benne, hogy még csak nem is tizenkettes számrendszerről van szó. Képtelen voltam szabadulni saját, egyedül lehetségesnek képzelt gondolati formáimtól" — emlékezik.

Süle Gábor is a rendszer alapszerkezetében várt különcebb gondolati sémákat: „Arra számítottam, hogy valami nagyon egzotikus vagy legalábbis bizarr számrendszer áll az idegen szavak mögött. Utólag azt gondolom: jóval többet lehetett volna bajlódni mondjuk a római számok vagy a francia számnevek álruhába bújtatott képviselőivel. Ezer (akarom mondani: tízezer) szerencse, hogy nem róluk volt szó."

Mégsem volt hiábavaló az ijesztgetés. Még így is előfordult, hogy valaki jól „dekódolta" ugyan a feladatban rejlő üzenetet, amikor azonban „kódolni" kellett volna, vagyis a felismert szabályszerűségeknek megfelelő „inverz transzformációkat" kellett volna alkalmazni, az már nem sikerült. Legtöbbször éppen a négyjegyű tagolás szokatlansága miatt.

A megfejtés

A legnagyobb körültekintéssel Dezső András oldotta meg a feladatot, az ő gondolatmenete alapján ismertetjük a megfejtést (programját mágneslemez mellékletünkön is közöljük).

„Az első információcsokor tanulmányozása után egy dologban lehettem csak biztos: mást jelent a SiShi, mást a ShiSi. Ebben a hitemben megerősített a harmadik kérdés is. A második csokorból először (6)-ra lettem figyelmes. Tudván, hogy Er nem nagyobb 10-nél, Yi-nek egynek vagy kettőnek kell lennie, különben a bal oldal nagyobb lenne tíznél, ez pedig ellentmondana a fenti állításnak. Tehát Yi = 1 és Er = 2, vagy Yi = 2 és Er = 6. Ezekkel az információkkal azonban tovább nem jutottam.

További megfigyeléseim során három csoportra osztottam a szócskákat. Az egyik fajta az, amelyik nem szerepel összetételekben. Ilyen a Yü, a Wan, a Qi és a Ling. Később kiderült, hogy a Qi másik csoportba tartozik, sőt a Ling is kakukkfióka. A másik fajta, amelyik szinte mindig csak összetételben található. Ezek a Qian, a Bai és a Shi. Őket szinte mindig megelőzi egy harmadik csoportba tartozó szó is (Yi, Er, San, Si, Wu, Liu, Qi, Ba, Jiu). A Shi különc: előtte is, utána is elhelyezkedhet a szócska, sőt állhat néha egyedül is. A Bai csak a Wan előtt magányos.

(10)-ből és (5)-ből felismertem, hogy Wan és Yü négyzetszámok, sőt mivel Yü a Wan-nak a négyzete, Yü negyedik hatvány. Lehet 16, 81..., de valószínűbb, hogy 10 000. Ennél nagyobb számra gondolni nem mertem. Reménykedni kezdtem, hátha mégis tízes alapú rendszerűek az ismeretlen nyelv számnevei.

(12)-ből kifejeztem Bai_Wan-t, (9)-ből Yü-t:

Bai_Wan = JiuShi_Jiu_Wan + Wan

Yü = JiuQian_JiuBai_Wan + Bai_Wan,

majd a második egyenletben behelyettesítettem Bai_Wan-t:

Yü = JiuQian_JiuBai_Wan +

+ JiuShi_Jiu_Wan + Wan.

Látszik, hogy ha a sok Jiu...-hoz hozzáadunk egy Wan-t, a Jiu-k sorra túlcsordulnak. A Jiu tehát a legnagyobb számjegy. Qian, Bai és Shi „kis" helyi értékek, Yü és Wan pedig „nagy" helyi értékek. Yü a Wan-nak a többszöröse, négyzete. Közöttük négy helyi érték különbség van. Ez azzal magyarázható, hogy Wan tízezer, Yü pedig százmillió. Qian így ezres, Bai százas, Shi tízes helyi értéket jelöl.

Feltevéseimet (15) is igazolni látszott. A számjegyek tehát nem hármas csoportokba rendeződnek, mint a magyarban, hanem négyesekbe. Ha a csoport után Yü vagy Wan áll, úgy a csoport számértékét szorozni kell százmillióval, illetve tízezerrel. Ha nincs utána semmi, akkor ez már a csoport számértéke. A csoporton belül a helyiértékszó határozza meg az adott helyi értéken lévő számjegyet. Ha a csoport középső helyi értékei közül valamelyik (vagy mindkettő) nulla, akkor egy Ling iktatódik be a nem nullák közé.

A számjegyeket már könnyebb volt megtalálni ezek után. (13) bal oldalán 90 + 90 áll, jobb oldala tehát 180. A tízes helyi értéken álló Ba tehát a nyolc, a százason álló Yi az egy. (6) miatt ekkor Er = 2. (11) jobb oldala 22, ezek szerint ShiYi tizenegyet jelent. A tizenvalahány úgy képződik, hogy a Shi után kerül a számjegy-szó. (4) bal oldali értéke 16, innen Liu = 6. (7)-ben egy szám négyzete meg egy egyenlő egy kerek számmal. Mivel a Shi nem önmagában áll, tehát a bal oldal nem tíz, ezért Qi nem lehet három, csak hét. (3) bal oldala 2 * 7 = 14, ebből Si = 4. (1)-ben valahányszor kettő egyenlő tízzel, Wu = 5. Már csak a három hiányzik: 19 bal oldalán 11 harmadik hatványa egyenlő 1331-gyel, San = 3.

Körülbelül ily módon tárulkozott elém az igazság. Ezek után a biztonság kedvéért ellenőriztem minden egyenletet: működött..."

D. A. gondolatmenetének empirikus igazolását jelentik az általa készített konvertáló programok. Van, aki rövidebb programmal oldja meg a konvertálást (például Csurgay Péter — ő nagyon ügyesen használta ki a rekurzióban rejlő lehetőségeket), D. A. programja azonban pontosabban kezel minden lehetőséget, és módszeres ellenőrzéssel kiszűri a felmerülhető hibákat. Igen alapos konvertáló programokat készített Süle Gábor is FoxPro 2.0-ben. Csaszny Márton részletes blokkdiagrammal írta le a megoldást, Domszky Zoltán pedig 14 lépésben foglalta össze a program működésének különböző állapotait: mikor mit kell vizsgálni, mit és milyen sorrendben kell megjegyezni, mit kell felírni, mi a foglalt tárolórekeszek tartalma. Mindegyiküknek gratulálunk, és sajnáljuk, hogy nem lehet tapasztalatcsereként valamennyiük megoldását közölni.

A megfejtéseket azonban nem mondjuk meg — tessék megkérdezni a programtól.

Vargha Dénes

Lapzárta után

A Novell cég szeptemberben ismét hasonló előadássorozatot tart, mint a júliusi Network University „Road Show" volt, s amelynek tapasztalatairól lapzártánk miatt csak a szeptemberi Alaplapban tudunk beszámolni. A téma ezúttal a hálózatos adatbázis-kezelők alkalmazása lesz. A Novell bevonja az oktatásba a Gupta Technologies és az Oracle szakembereit is.

Valami azért marad belőle

Volt egyszer egy PC Turbo Klub

Bizonyára többeknek feltűnt, hogy a PC Turbo Klubba invitáló válasz-levelezőlap szokásos helyét mostani számunkban valami más foglalta el. Ez a „valami más" eddigi koncepciónk módosítását jelenti. A PC Turbo Klub funkcióját, működési formáját átalakítjuk.

Az 1991 elején indított speciális előfizetési forma olvasóink körében gyorsan népszerűvé vált. A számonkénti 20 forintos engedmény és az áruházi vásárlási kedvezmény mellett sokan fogadták örömmel a szakmai bemutatókat, a konzultációs lehetőségeket, az információszolgáltatást. Végtére is nem rossz dolog olyan klub tagjának lenni, amely csak szolgáltatásokat és kedvezményeket nyújt, a klubtagokat pedig semmilyen kötelezettség nem terheli.

Anélkül, hogy financiális részletekkel terhelnénk olvasóinkat, el kell mondanunk azt, ami mostanra számunkra egyértelművé vált: a klub bevételi forrásai mára már nem fedezik azokat a többletráfordításokat, amelyeket a rendezvények szervezése, a nyilvántartás adminisztrációja, az expediálás, a reklamációk intézése jelent, nem is szólva az emelkedő postai tarifákról.

A mi esetünk tanulságos lehet a hasonló cipőben járó többi szerkesztőség számára. A lap fennmaradása érdekében minimális létszámmal dolgozó szerkesztőségi és kiadói gárda ne törekedjen olyan terjesztési funkciók átvállalására, amelyek hatékonyabban elvégeztethetők a terjesztésre szakosodott nagyobb szervezetekkel. Mi legalábbis erőnkön felüli feladatra vállalkoztunk a klub működtetésével, és a rendezvények szervezésére vagy külön szolgáltatások nyújtására nem jutott elegendő energiánk, a lap készítése túlságosan leterheli 4 fős főállású szerkesztőségi gárdánkat.

Aki eddig jutott az olvasásban, nyilván úgy gondolja: „Na, kész, ezek is becsukják a kaput, nincs tovább klub, nincsenek kedvezmények, vegyem meg az Alaplapot az újságárusnál vagy fizessem elő a Postánál." Ami az előfizetést illeti, valóban nincs más lehetőség, mint visszatérni a Postához. Ez a váltás anyagilag számunkra is bizonyos veszteséget okoz, mégis vállalnunk kell, hogy minden erőnket a lap jobbítására fordíthassuk. Az igazsághoz tartozik még, hogy az előfizetők többsége a PC Turbo Klub létrehozása után is kitartott a hagyományos postai előfizetési mód mellett. (Szerencsére!)

Azt is be kell vallanunk azonban, hogy egy kicsit fáj a szívünk ezért a PC Turbo Klubért. Valami hasznos és jó dolgot kezdeményeztünk, ami által személy szerint is közelebb kerültünk lelkes és hűséges olvasóinkhoz. Sokáig törtük a fejünket, hogyan lehetne megmenteni valamit ebből a klubból. Arra a következtetésre jutottunk, hogy nagyobb munkaráfordítás nélkül megőrizhetjük — sőt kiterjeszthetjük — szakmai közvetítő szerepét, ha a tagok címét adatbázisba szervezzük. Úgy véljük, hogy a címanyagot felhasználó cégektől olvasóink olyan információkat is megkaphatnak, amelyek közlésére a lapban nincs lehetőség. (Például helyi és regionális rendezvényekre szóló meghívók.)

Ez a lista akár a teljes olvasótábort felölelheti. Úgy képzeltük el, hogy mindenki eldönti, akar-e szerepelni e megváltozott funkciójú klub nyilvántartásában. Aki igen, az mindösszе annyit vállal, hogy címét megadjuk az értékesítési, konferenciaszervezési vagy más szakmai céllal címanyagot kereső cégeknek, szervezeteknek. A Cédrus Karolina Áruház és a Floppyland Kft az adatbázisban szereplőknek pedig továbbra is 5%-os vásárlási kedvezményt ad. Az eddigi PC Turbo Klubtagok automatikusan bekerülnek a címadatbázisba, hacsak a kartonból kivágható válasz-levelezőlapon vagy más módon nem kérik ennek ellenkezőjét. Aki nem volt tagja a PC Turbo Klubnak, az viszont kérheti, hogy neve és címe szerepeljen ebben az új listában. A válaszadásra nem szabunk határidőt, mert a klub teljesen nyitott, bárki bármikor kérheti a címanyagba való felvételét vagy onnan való törlését.

Végül pedig alapvető szolgáltatásunkat illetően a „régi" klubtagok mindaddig továbbra is közvetlenül tőlünk kapják az Alaplapot, amíg „le nem cseng" az előfizetésük. Az előfizetések meghosszabbítása azonban már csak a Postánál lehetséges.

Varga János

Mikrogépgyártás

Amerikai számítógépgyártók egy csoportja a mikroszámítógépek előállítási adatainak nyilvántartására alakított bizottságot. Japánban már működik hasonló szervezet, amely az utóbbi tíz évre gyűjt iparági adatokat, Nyugat-Európában pedig most szerveznek egy ilyen csoportot. Az amerikai tömörülés munkájában többek közt részt vesz az IBM, az Apple és a Compaq. A bizottság negyedévente kívánja ellátni tagjait a mikroszámítógépek gyártására vonatkozó adatokkal, vázolva a fontosabb trendeket.

Új könyvtár a BME-n

Mintegy negyvenmillió forint felhasználásával korszerűsítették a Budapesti Műszaki Egyetem könyvtárát. Ez nem elsősorban az új olvasótér kialakítását jelenti, hanem az integrált számítógépes rendszer bevezetését. Lelkét az izraeli Ex Libris cég által kifejlesztett és szállított Aleph nevű szoftver jelenti. Ez kifejezetten egyetemi környezetben készült, hasonló működik Európa negyvenöt helyén, többek között a dán műszaki egyetemen, a genfi nukleáris kutatóközpontban, valamint New Yersey Princeton egyetemén. Izraelben minden egyetem ezt a rendszert használja. Hasonló célt tűztek ki Magyarországon is.

Judy

Elkészült a Judy szoftvertervező és programíró rendszer új verziója, mely már Turbo Pascal 6.0 forráskódú programot is ír. Alkalmazásával a felhasználói programok széles körét elegendő csupán — interaktív üzemmódban — vázlatosan megtervezni, s a rendszer másodpercek alatt megírja a forrásnyelvű programot. Az így kapott eredmény jellemzői: a Turbo Pascal választott verziójában szintaktikailag hibátlan, strukturált, futási időre, tárra optimalizált programok keletkeznek. A Judy által írt programok képesek más, forrásnyelvű fájlokból eljárásokat, függvényeket beszúrni, használni éppúgy, mint lefordított (.EXE vagy .COM) programokat meghívni. A létrehozott programok más programnyelvekben — például Clipper, C, Cobol — is előnyösen használhatók. A kész programok később akárhányszor módosíthatók. A rendszer megvizsgálja a hardvert, ezt a tulajdonságot a generált programok automatikusan öröklik, azaz öninstallálókká válnak. Így például automatikusan generálódnak az egérkezelési funkciók a leendő programba. A létrehozott programok .PAS kiterjesztésű fájlokba kerülnek, s éppen olyanok, mintha szövegszerkesztővel írtuk volna őket. A Judy már jelenleg is tantervi anyag a Kandó Kálmán Műszaki Főiskolán.

Számítástechnika farmereknek

A mezőgazdasági kisüzemek számítástechnikai hátterének megteremtését tűzte ki célul az Agroorg Kft. A tapasztalatok szerint ugyanis napjainkban már a hazai mezőgazdasági közép- és kisüzemek sem nélkülözhetik azt a biztonságot, amelyet külföldön a korszerű számítástechnika nyújt a termelőknek. E háttér hazai kialakításának megalapozásaként kötöttek szerződést a Siemens Ausztria céggel. Ennek értelmében megkezdik az osztrák cég által kifejlesztett, s a farmergazdaságokban jól hasznosítható programrendszer forgalmazását. Ez egy számítógéppel támogatott termelésirányítási és -elszámolási rendszert foglal magában, amely felhasználható többek között a növénytermesztés és az állattenyésztés irányítására, és alkalmas az ügyviteli munka, többek között a főkönyvi könyvelés elvégzésére.

Számítógép-sűrűségi világcsúcs

Svájcban 1990-ben az állami hivatalokban és a gazdaságban alkalmazott számítógépek száma 130 ezerrel nőtt, s elérte az 530 ezret. Igazság szerint 156 ezer számítógépet helyeztek üzembe, de ezek közül 20 ezer a kiöregedett berendezések pótlására szolgált. 1990 végén több mint ötezer üzemben 17 ezer számítógépes munkahely üzemelt, 6500-zal több, mint az előző évben. Svájcban már 1990-ben is 1000 munkavállalóra 180 mikrogép jutott. Ezzel Svájc megszerezte a világelsőséget a számítógép-sűrűség tekintetében.

Jóslat az ezredfordulóra

2000-re a fejlett tőkés országokban a foglalkoztatottak negyven százaléka számítógépes munkahelyen fog dolgozni, és minden ötödik munkavállalótól elvárják a főnökei, hogy kreatív módon kezelje a gépet, azaz az egyszerű szoftverhasználatnál többre is képes legyen. Ez a jóslat hangzott el a tekintélyes osztrák Der Standard című folyóiratban. A fejlődés azonban korántsem ennyire egyértelmű. A számítógépiparban ugyanis a fejlődés túlságosan gyors, az emberek tömegei egyszerűen nincsenek felkészülve arra, hogy az egyre bonyolultabb információrendszerekben eligazodjanak, annak összes lehetséges következményeit és hatásait felmérjék. A fejlődés geográfiai motorjaként elismert japánok és amerikaiak mellett immár csaknem bizonyosra vehető, hogy az európaiakra csupán a néző szerepe marad.

Számítógép a bűnüldözésben

A BM és az SZKI Pixel együttműködésével kifejlesztett rendszer a számítógép mesterséges intelligenciáját hívja segítségül a tettesek felkutatásához. A lényege, hogy — a hagyományos mozaikkép-készítési eljárással szemben — a tanú által kiválasztott, a tettesre leginkább emlékeztető arcrészletet elképesztő valósághűséggel illeszti a monitoron lévő arcba. A program bármilyen képtranszformációs műveletet elvégez, nagyítja, kicsinyíti, árnyékolja, ráncosítja, tetszés szerint. Az így kapott portré fényképszerű ábrázolást ad a tettesekről. Közel tíz rendőrkapitányságon használják már a képfeldolgozó rendszert, méghozzá sikerrel: jó néhány veszedelmes bűnöző akadt horogra az ekképpen készült portrék segítségével.

Hasonlóképpen a bűnüldözés segítségére lehet a BM és a Geometria együttműködésével kifejlesztett térinformatikai rendszer is. Ez tematikai szempontok szerint képes megjeleníteni az adott városrész épületeinek alaprajzát, csatorna- vagy elektromos hálózatát, megmutatva például a bűnöző lehetséges menekülési útvonalait. Ha a számítógépbe beviszik, hogy hol, mikor, miféle bűncselekményeket követtek el a városban, akkor a rendszer — megkeresve az összefüggéseket — tájékoztatja a rendőröket a város rablási szempontból legfertőzöttebb pontjairól. Ehhez igazodva aztán a rendőrség hatékonyabban tervezheti meg például a járőrszolgálatok útvonalait.

Végezetül essék még szó egy, szintén a mesterséges intelligenciára épülő, ám ma még igencsak a tervezés állapotában lévő ötletről. Az ORFK a Számalk Multilogic Kft.-vel együtt gondolkodva egy olyan, nyomozást támogató, szakértői rendszer kidolgozásán fáradozik, amely segítséget nyújthat az egyes bűncselekmények felderítéséhez. Számítógéppel feldolgozva például a nyomozóknak a helyszíni szemlékkel kapcsolatos évtizedes tapasztalatait összevethetik az éppen aktuális helyszíni szemlén összegyűjtött információkkal, a gép pedig lehetséges alternatívákat adhat a nyomozás elindításához.

A Mikrobazár rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk.

A kereskedelmi tevékenységet szolgáló apróhirdetések tarifája gépelt soronként (60 karakterenként) 300 Ft. Kérjük, hogy a hirdetés díját a Cédrus Kiadó Kft-nek az Általános Értékforgalmi Banknál vezetett 204-19417 számú számlájára utalják át, vagy postautalványon fizessék be a Cédrus Kiadó Kft címére (1441 Budapest VIII., Reguly Antal u. 8.), a hátoldalon feltüntetve, hogy apróhirdetés. A befizetést igazoló szelvényt a közlendő hirdetési szöveggel együtt az Alaplap szerkesztőségéhez (a kiadóéval azonos címre) küldjék el.

A nem kereskedelmi célú egyéni hirdetések közlése INGYENES!

ELAD

Eladó **Commodore 64**-es gép magnóval, kazettákkal, joystickkel, lemezekkel. Ára: 13 000 Ft. PC-s programcsere is érdekel. Cserealap: Street 1-2, teljes Sim-sorozat, Flight Sim V3, stb... Keresem a Jones és a Lemmings elnevezésű programokat. Cím: Bozsó G. István, 7400 Kaposvár, Kisközi u. 18.

3 éves **Commodore 64**-es gép olcsón eladó. Ára: 9991 Ft. Cím: Jozefák János, 1181 Budapest XVIII., Háromszék u. 15.

Eladók **Commodore 64**-hez az alábbiak: 1802 színes monitor, 1541 floppy, 1535 datenrecorder, Final Cartridge III., valamint 10 db Data Becker könyv. Ára együttesen 26 000 Ft. Cím: Kun József, 1213 Budapest XXI., Puli sétány 19. III./14. Tel.: 276-7225.

Eladó **Sinclair Spectrum** számítógép összes tartozékával, 128 kB memóriával. Cím: Dietrich András, 8700 Marcali, Széchenyi u. 16. IV./13.

Eladó **IBM XT**: 640 kB RAM, 20 MB winchester, 360 kB floppy, Hercules kártya, 14"-os monokróm monitor, 101 gombos billentyűzet. Ára: 25 000 Ft. Cím: Oláh Endre, 1111 Budapest XI., Irinyi J. u. 9-11. 866-os szoba (este 18-20-ig).

Eladó **AT 286**: 1 MB RAM, 1,2 MB floppy, monokróm monitor, 101 gombos billentyűzet. Cím: Lovas Tibor, 7030 Paks, Bercsényi u. 46. Tel.: (75) 14-313.

Eladók **programleírások** Commodore, Amiga, Spectrum és IBM gépekhez. Válaszboríték ellenében leírást küldök a kiválasztott programról. Cím: Heszler Róbert, 8220 Balatonalmádi, Móra F. u. 5. II./8.

PC video vezérlőkártyák (Hercules, CGA, EGA, VGA) hardveres ékezetesítése CWI vagy egyéb tetszőleges kódkészlet szerint. PC Turbo Klubtagoknak kedvezmény! Cím: Lóth Tamás, 1558 Budapest, Pf. 76.

Számítástechnikai oktatás IBM PC gépen bármilyen témában a legolcsóbban! Beszerzési tanácsadást és programkészítést is vállalok! PC Turbo Klub-tagoknak kedvezmény! Cím: Fridl György, 1116 Budapest XI., Szalóki u. 6.

K—ÉP Stúdió **építészeket** keres CAD-szolgáltatói és kereskedelmi-ügynöki feladatokhoz. 5-20%-os jutalék! Cím: K—ÉP Stúdió, 1426 Budapest, Pf. 66. Tel.: 133-4760/328-as mellék (Kuczogi László).

Objektumorientált programozás Clipper 5.01-ben: Objects 1.1. Kérésre tájékoztatót küldök. Cím: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. IV./13. Tel.: (42) 13-568.

Programokra, segítségre, cseretársakra van szükséged? Az ASIS megoldja problémáidat! Bárhol laksz, bármilyen géped van, írj! Kérésre ingyenes tájékoztatót küldünk. Cím: ASIS, 1425 Budapest, Pf. 729. Tel.: 142-8075.

VESZ

Vennék jelképes áron használható állapotban levő ZX Spectrum gépet tartozékokkal és magyar nyelvű leírással. Cím: Helfrich Zoltán, 1062 Budapest VI., Bulcsu u. 8.

CSERÉL

Amigára zenemodulokat, **Roland MC-50**-re songokat, **MSX**-re programokat cserélek vagy adok. Cím: Jancsurák István, 3528 Miskolc, Dráva u. 7. Tel.: (46) 380-251.

PC-n friss **stuffokat** cserélünk, valamint alakuló csapatunkba tagokat keresünk: grafikus, zenész, swapper, assemblyben programozó munkatársak jelentkezését várjuk. Ugyanitt Adlib hangkártya 20 000 forintért eladó. Cím: Fábián Zoltán, 8500 Pápa, Martinovics u. 9.

Cserélnék **IBM XT/AT** programokat. Válaszboríték ellenében listát küldök. Ugyanitt eladó ZX-81 tartozékokkal. Ára: 3000 Ft. Megrendelhető továbbá IBM AT-hoz soros hangadapter. Ára: 800 Ft. Cím: Grexa Miklós, 2628 Szob, József Attila út 3.

Bibliográfia

A hónap témájához az 1991/4. számunkban megjelent összeállításunkat ajánljuk olvasóink figyelmébe. Most egy 13 kötetesre tervezett sorozat eddig megjelent köteteinek adatait ismertetjük. A sorozat kiadója a Miskolci Egyetem Dunaújvárosi Főiskolai Kar Kiadói Hivatal.

Dr. Fercsik János: Az IBM számítógép használata 1. (A gép — Programok — A gép be- és kikapcsolása — DOS parancsok — Program beolvasása, indítása — A KEYDEF program — A PCTOOLS program) Dunaújváros, 1992. 152 oldal. Ára: 200,— Ft.

Dr. Fercsik János: Az IBM számítógép használata 2. (Szövegszerkesztés, a ChiWriter programcsomag) Dunaújváros, 1992. 100 oldal. Ára: 190,— Ft.

Dr. Fercsik János: Az IBM számítógép használata 3. (Táblázatkezelés, a Lotus 1.2.3. programcsomag) Dunaújváros, 1992. 151 oldal. Ára: 200,— Ft.

Dr. Fercsik János: Az IBM számítógép használata 4. (Tervezés, szerkesztés: az AutoCAD programcsomag) Dunaújváros, 1992. 276 oldal. Ára: 375,— Ft.

Dr. Fercsik János: Az IBM számítógép használata 5. (Profi billentyűzetdefiniálás: KEYBPPAT — Adatállományok kezelése: dBASE III.PLUS, CLIPPER) Dunaújváros, 1992. 232 oldal. Ára: 290,— Ft.

Dr. Fercsik János: Az IBM számítógép használata 6. (Nyomtatás: az EPSON FX 1050 nyomtató — A NORTON programok: Commander, Integrator, Editor, Guide — A SIDEKICK titkár(nő) program — az IBM PC gépek helyi hálózatban: a NOVELL hálózat — Vírusok, víruskutatás. Szérumok, vírusölés) Dunaújváros, 1992. 207 oldal. Ára: 255,— Ft.

Dr. Fercsik János: Az IBM számítógép használata 7. (Programozás BASIC nyelven: a GWBASIC program) Dunaújváros, 1992. 200 oldal. Ára: 255,— Ft.

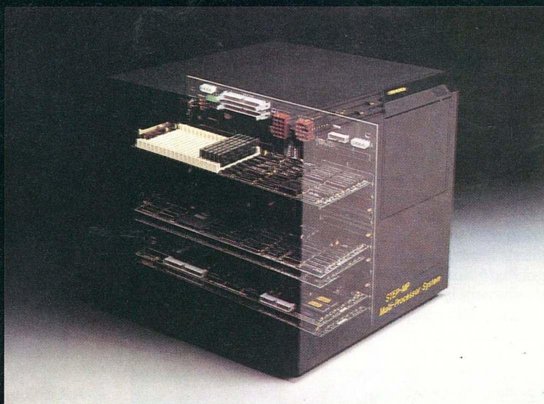
Dr. Fercsik János: Az IBM számítógép használata 8. (Programozás PASCAL nyelven: a Turbo-PASCAL program) Dunaújváros, 1992. 338 oldal. Ára: 430,— Ft.

Dr. Fercsik János: Az IBM számítógép használata 9. (Kiadványszerkesztés: a Ventura HVP program — Képek, rajzok, szövegek, programok beolvasása: scanner) Dunaújváros, 1992. 304 oldal. Ára: 405,— Ft.



INTRAM Szerviz és Kereskedelmi Kft.

1072 Budapest VII., Kis Diófa u. 6.
Telefon: 122-0087 Fax: 121-3230



Ilyen még nem volt Magyarországon!

Everex és Wyse számítógépek a profiknak, akiknek csak a legjobb elég jó

Olcsó amerikai számítógépek azoknak, akiknek számít, mit kapnak a pénzükért

Sysdoki és Sysguard mindenkinek, akinek fontos a vírus- és adatvédelem

Érték ügyviteli- és vezetői információs rendszer
azoknak, akik tudni akarják, hogyan áll a cégük

Aprócikkek, meglepetések, kedvezmények mindenkinek, aki szeret jót vásárolni

Nálunk a minőség mindig megéri az árát!

INFORMÁCIÓKÉRÉS: 35 ▼

Intelligencia minden szinten

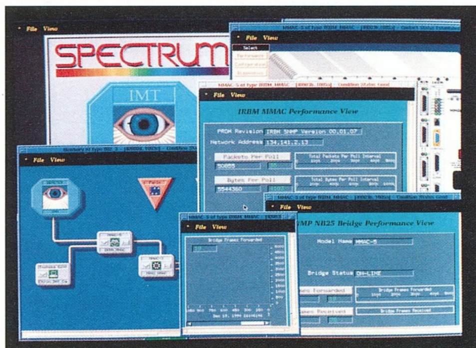
E havi összeállításunk egymástól igencsak távol eső termék- és szolgáltatáskínálatot vonultat fel — a hálózatépítési módszerektől kezdve a COCOM-rendelkezések könnyítésén át az adatvédelemmel és a mesterséges intelligenciával kapcsolatos hazai kutatás kimagasló eredményeinek bemutatásáig.
A közös vonás, amely alapján rokonítható e négy terület: a potenciális felhasználó kényelmének szolgálata.

Válás után újabb frigy

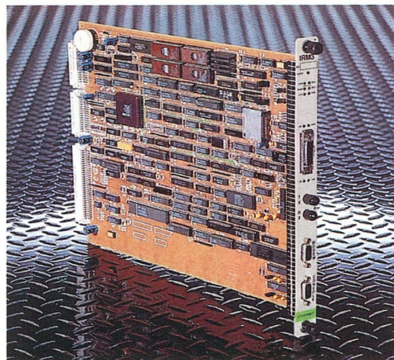
Alig egy éve számoltunk be arról, hogy a hálózati termékek forgalmazása területén a Rolitron megállapodást kötött a BICC-vel. A házasság azonban rövid életűnek bizonyult. Ugyanis a pénzügyi nehézségekkel küszködő — de kiváló technikai képességekkel rendelkező — BICC-t felvásároló 3Com filozófiája alaposan eltér a rendszerintegrátor Rolitron koncepciójától.

A Rolitron újonnan választott partnere erős technikai háttérrel és komplex termékskálával rendelkezik. A Cabletron a világon a harmadik helyen áll (a Synoptics és az Ungerman Brass után) a hálózati elemek forgalmazásában. A Cabletron-termékek a hálózatokkal kapcsolatos valamennyi igényt kielégítik. A Cabletron tetszőleges számú számítógép között létesít hálózati kapcsolatot, legyenek azok a világ bármely részén. Egyaránt vállalkoznak Ethernet, Token Ring és FDDI lokális és nagy távolságú hálózatok kiépítésére. Választékukban teljes a hálózati kapcsolóelemek és szerelési anyagok repertoárja: kábelek, transceiverek, menedzselhető hálózati csatolók, intelligens moduláris felépítésű hálózati elemek (hubok, repeaterek, lokális és remote bridge-ek) mellett az alkatrészek, kiegészítők és tesztműszerek alkotják a Cabletron termékpalettáját.

Tekintettel arra, hogy a Cabletron mindent maga gyárt — megbízható minőségben és alacsony áron —, az értékesítéskor nincsenek beszállítási problémák, és rövidek a szállítási határidők.



A Cabletron-termékek szabványosak. Kezdetben csak követték a szabványokat, ma már viszont ők írják azokat. A velük kapcsolatban álló cégek (Novell, Banyan...) úgy alakítják termékeiket, hogy azok illeszkedjenek a Cabletron-palettára. A gyártás az Integrált Hálózati Architektúrán alapul (INA). Nagy segítséget jelent a felhasználóknak, hogy a Cabletron-termékek nemcsak egymással, hanem más cégek termékeivel is kompatibilisak. Így globális, osztott hálózatok építhetők ki és irányíthatók olyan termékekkel, amelyeket hálózati szabványtól függetlenül egy közös rendszerben működtetnek együtt.



A Cabletron háromféle hálózatirányítási eszközt is kínál (MMAC, Remote Lanview, Spectrum). A hardveralapú, moduláris Multi Media Access Center (MMAC) az Ethernet és FDDI (üvegszálas) hálózati elemek mellett Token Ring szegmenseket is képes hálózatba egyesíteni. A Remote Lanview olyan hálózatirányítási szoftver, amely figyeli és ellenőrzi a hálózati kapcsolatokat, valamint diagnosztizálja a felmerülő problémákat. A Unix-alapú Spectrum pedig a különféle hálózati topológiák támogatása mellett figyeli és ellenőrzi a különböző gyártók termékeiből kialakított heterogén hálózatokat.

A Rolitron komoly tapasztalatokat szerzett a hálózatépítés területén. A Cabletronnal kötött VAR-szerződés értelmében elsősorban az Ethernet, Token Ring és Apple Talk lokális hálózatok építéséhez szükséges elemek teljes választékával és értékhozzáadó tevékenységével (oktatás, műszaki támogatás) igyekszik valóban működő hálózatokat kiépíteni.

Szigorúan ellenőrzött komputerek...

A hazai számítástechnika fejlődésére erőteljesen rányomta a bélyegét, hogy az igazán korszerű, nagy tudású gépek vásárlásakor meg kellett várni a COCOM-engedélyt. Ez gyakran akár fél évbe is beletelt. A reménybeli felhasználó és a hazai forgalmazó egyaránt türelmetlenül várta az engédélyt. Tudjuk, hogy sokszor ugyanaz a gép vándorolt kiállításról kiállításra, termékbemutatóra. S mivel csak egyetlen darab ilyen számítógépük volt, a jó paraméterekkel rendelkező gép hazai piaci bevezetésénél ez az eljárás nem jelentett előnyt.

Ezen a (magyar felhasználónak) áldatlannak tűnő helyzeten segített 1990-ben egy kormányrendelet, amely lehetővé tette, hogy állami garancia segítségével, leegyszerűsített módon kerüljenek a fejlett számítástechnikai eszközök Magyarországra. Így például az IBM az elsők között kapott exportengedélyt Magyarországra a lebegőpontos processzoros gépek, RISC-rendszerek, adatkódoló berendezések, valamint ezek operációs rendszereinek és felhasználói szoftvereinek importjára.

A helyzet tavasz óta tovább javult. Ugyanis az IBM Magyarország Kft. a saját és az IBM világszervezetének garanciájával — a szuperkomputerek és a kriptográfiai termékek kivételével — viszonteladói keretengedélye terhére bármilyen számítástechnikai eszközt szabadon importálhat, betartva a behozatali eljárásokat.

A magyarországi IBM naprakész információkkal rendelkezik a belföldön eladott, üzembe helyezett termékekről és technológiákról. Ugyanis az ellenőrzött termékek nyomkövetését a COCOM-tagországok továbbra is evidenciában tartják. További könnyítés, hogy a szupergépek és a kriptográfiai eszközök esetében — a vevő nevében — a magyarországi IBM kéri meg az egyedi engedélyt az exportáló országtól.

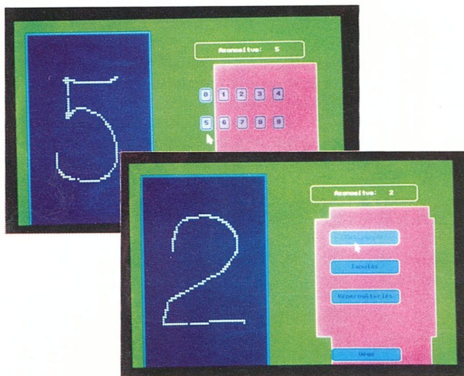
Az új helyzet kínálta lehetőségek a továbbiakban nem korlátozzák a legkorszerűbb technológia magyarországi beáramlását, amely remélhetőleg kedvező hatással lesz a hazai számítógéppark fejlesztésére.

Hogyan szimuláljunk? — Neuronhálózattal!

Az emberi agy felépítését és működését utánozó neuronhálózatok olyankor segíthetnek, amikor a feladatok megoldásának pontos algoritmusát nagyon nehéz vagy lehetetlen meghatározni. Ilyen alkalmazási terület például a beszéd- és alakfelismerés vagy a képfeldolgozás. A mesterséges neuronhálózatok — az emberhez hasonlóan — minták alapján tanulnak, ezeket memorizálják, sőt a minták alapján még általánosítani is tudnak.

Alkalmazásuk területén néhány új eredmény született szerte a nagyvilágban. Neuronhálózat dolgozik a manapság oly népszerű pen-computerben a ceruzával képernyőre írt szöveg értelmezésekor; a robotok, amelyek kezelőjük szavát megértik és végrehajtják, szintén ezen az elven „gondolkodnak". Ugyancsak neuronhálózat működik abban a repülőtéri biztonsági berendezésben, amely megtalálja a csomagokba rejtett robbanóanyagot, vagy — a repülés világánál maradva — a radar- és infravörös képek alapján a repülőgépet tetszőleges helyzetből és irányból felismerő berendezés belsejében is.

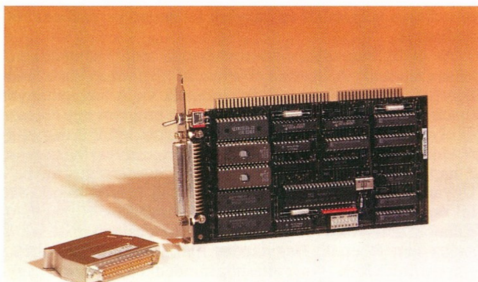
Az eddig használatos neuronhálózati modellek rendkívül lassan tanultak. A Gerenia elnevezésű neuronhálózat-szimulátort magyar szoftveresek fejlesztették ki. A hagyományos (back propagation) algoritmus helyett a program egy teljesen új elven (Classitron) alapul. Így a tanulási idő 2-3 nagyságrenddel(!) csökken. Például a kézzel írt számok felismerése 3 nap helyett 10 perc alatt beépül a „mesterséges agyba". Ugyanez kézzel írt betűk esetében 2 hét helyett mindössze fél órát vesz igénybe. Így olyan alkalmazások is megvalósíthatók, amelyek a rendkívül hosszú tanulási idő miatt eddig szóba sem jöhettek. Hagyományosan — a kellő gyorsaság elérése érdekében — a tanításhoz mindenképp célszerű nagy- vagy szuperszámítógépet használni, még ha az applikáció PC-n is fut majd, míg a Classitron módszerrel a tanításhoz is elég a PC.



A Gerenia olyan fejlesztőrendszer, amellyel rendkívül gyorsan készíthetők alkalmazások. Az Aktív Rekord Bt. az Ifabón mutatott be egy alkalmazást (kézzel írt számok felismerését programmal); ez 48 óra alatt készült el, beleértve a hálózat betanítását is. (A demóverzió a lemezmellékleten megtalálható.) Ugyanez a feladat hagyományosan hetek--hónapok munkáját jelentette volna. Nem véletlen, hogy a külföldi szaksajtó (Computer Product News) is felfigyelt a magyar szenzációra, és májusi számában már beszámolt a szakirodalomban jelenleg egyedülálló gyorsaságú neuronhálózatos rendszerről. S ha hozzátesszük, hogy a Turbo Pascalban írt fejlesztőrendszer nem is drága (99 000 Ft), valamint hogy az alkalmazások készítése rendkívüli módon lerövidül, akkor nem kérdőjelezhető meg az állítás: a már említett felhasználási területek mellett például az EKG-analízisnél vagy pénzügyi, gazdasági előrejelzéseknél lehetne okosan használni a mesterséges intelligencia e hazai gyöngyszemét.

Hogyan védekezzünk? — Titkosan!

Mindenki, aki számítógépen dolgozik, valósággal retteg attól, hogy adatai illetéktelen kezekbe kerülhetnek. Hiszen — nagy



általánosságban — jelenleg bárki bejuthat egy hálózatba egy PC-n keresztül, s ott nemcsak a vírusok terjedését idézheti elő, hanem mások féltve őrzött adatai között is szabadon garázdálkodhat. De ha az előre megfontolt szándékkal kalózkodó „okos számítástechnikusokat" figyelmen kívül is hagyjuk, gondolni kell azokra a nemtörődöm laikusokra is, akik gondatlanságból például kidobálják a bérlistákat, s ezeket a későbbiekben bárki tanulmányozhatja. Emlékezzünk csak arra, milyen sokszor csomagoltak a piacon valami portékát olyan papírba (listába), amelyen egy vállalat dolgozóinak pontos címe, összes adata és fizetése szerepelt. Ugyancsak problémák forrása lehet például, hogy rossz adatátviteli környezetben a faxszolgáltatások is illetéktelen kezekbe kerülhetnek.

Adatvédelmi megoldásra egymástól függetlenül két cég is kínál hasonló, de más fajsúlyú terméket. Mindkét társaság olyan kártyát fejlesztett ki, amely — külön-külön, az adó és a vevő gépébe helyezve — a számítógépben és a hálózatokban őrzött adatok és információk magas szintű védelmét biztosítja. A kártyával rejtjelezik a diszken tárolt adatokat, visszakereséskor pedig időlassulás nélkül dekódolják azokat. Így a felhasználó ebből a procedúrából semmit sem vesz észre. Minden kezelőnél más és más a rejtjelezési kulcs, így mindenki csak a maga által tárolt adatokhoz férhet hozzá. De természetesen vannak olyan általános kulcsok is, amelyek egy bizonyos adatkörre vonatkozó információkat rejtjeleznek. A kriptográfiai rendszer felhasználói így biztosak lehetnek abban, hogy a titkosan tárolt vagy közvetített üzeneteket mások nem tudják megfejteni, hamisítani, illetve manipulálni.

Az ITEA Kft. által fejlesztett rendszerben egy komplett szett kell ahhoz, hogy valaki hozzáférjen a számítógéphez. Mint hardver, ez egy kriptográfiai alapkártya (CryptoPCard), s a központi kulcsellátó berendezés, valamint egy külső eszköz jelenlétét feltételezi. A CryptoPCard tulajdonképpen szabványos AT bővítőkártya, amelyet a központi kulcsellátó berendezés lát el rejtjelezési kulcsokkal. A kulcsgenerálást a felhasználó maga is elvégezheti, mivel a CryptoPCard az alkalmazó saját rejtjelezési algoritmusát is értelmezi. A kulcs és az algoritmus egy része a külső eszközben található. A rejtjelezés visszafejtésére nincs mód, hiszen ehhez valamenynyi fizikai és logikai elemet kellene egyidejűleg ismerni.

A rendszerben az univerzálisan kialakított moduláris szoftver a felhasználói programokkal tehető egyedivé. Adatvédelmi rendszer alkalmazása esetén a felhasználói programokat és az adatbázisokat kriptográfiai szempontból is meg kell tervezni. Ilyenkor kell dönteni a használandó kulcsrendszerről, a hozzáférési hierarchiáról, a fájl- vagy mezőszintű titkosításról és a rejtjelezés algoritmusáról. De nem okoz gondot az sem, hogy a már működő programokat kriptoparancsokkal egészítsük ki. Az államigazgatásban, az önkormányzatoknál, a banki szférában, a biztosítóknál és az egészségügyben kitűnően használható kártya 45 000 és 79 000 Ft közötti (kiépítéstől függő) árával méltányos beruházás az adatok védelme érdekében.

Az olcsóbb árfekvésű, de felhasználóbarát és már XT-ben is használható — az előbbivel ennyiben konkuráló — kártyát (BIS SEC 1.0) a Bis Kft. fejlesztette ki. Ez néhány hasonló szolgáltatásával 50 000 forintért kínál alternatívát a felhasználók egy részének.

A rejtjelezés bájtról bájtra halad, az adatok a processzorban nem olvashatók ki, a kódolás-dekódolás kívülről nem követhető. A fájl- és adatrejtjelezést megvalósító kártyán valódi véletlenszám-generátor van. Fontos, hogy rendszeresen ellenőrzik az öntesztek működését. A hálózatban is működő rendszerben a beépített hardverérzékelő (egy floppy) erősíti a PC fizikai hozzáférés-védelmét. Ha kivesszük a 3,5"-os floppyt a gépből, akkor elsötétül a képernyő, amellyel a hozzáférést és a számítógépet egyaránt védjük. Az adatvédelmi rendszer kezelése kényelmes, hiszen a 3,5"-os floppyval ugyanazt csinálhatjuk meg, mint jelszóval és egyéb segédeszközökkel.

Az adatvédelem egyre növekvő fontosságát jól jelzi, hogy az Országgyűlés is tárgyalja az adatvédelmi törvényt. A „klasszikus" és a profi felhasználók kellemes helyzetben vannak, mert eldönthetik, hogy a különböző „védekezési módok" közül melyiket válasszák.

Sziebig Andrea



RC dtp-Fotó ARTÓ

ComputerLand®

1055 BUDAPEST BALASSI BÁLINT U. 7.
TELEFON: 269-0171 • FAX: 269-0178